FREE DEMO SAMPLE

Guessing Paper 2026-27 Demo Set

BCA · Semester 3 · Relational Database Management System (RDBMS)

ABVV Bilaspur — अटल बिहारी वाजपेयी विश्वविद्यालय, बिलासपुर

यह सिर्फ़ एक डेमो सेट है 📘

इस डेमो से आप हमारे उत्तरों की क्वालिटी और पेपर का फ़ॉर्मेट देख सकते हैं। **असली 3 पूरे Guessing Paper सेट** (Set A, B, C) सिर्फ़ **Pariksha Pass App** में मिलते हैं:

- हर विषय के **3 पूरे सेट** — Most Probable, Highly Likely और Alternate
- हर प्रश्न का **सटीक मॉडल उत्तर** — हिंदी और English दोनों में
- **Year Tag** — जो प्रश्न पहले यूनिवर्सिटी परीक्षा में आ चुके हैं
- आपकी **यूनिवर्सिटी के पैटर्न** पर — पूरा सिलेबस कवर

▶ Pariksha Pass App डाउनलोड करें
यहाँ टैप करें — Google Play पर उपलब्ध

QR स्कैन करें

इस डेमो के प्रश्न असली Set A/B/C से नहीं लिए गए हैं। कोई भी पेपर परीक्षा में प्रश्न आने की गारंटी नहीं देता — इसे पाठ्यपुस्तक के साथ अभ्यास के लिए उपयोग करें। · Free sample shared on cguniversities.com

Paper Code: **CASC-08T**
DEMO SET

Roll No. | | | | | | | | | |

BILASPUR UNIVERSITY, BILASPUR (C.G.)
बिलासपुर विश्वविद्यालय, बिलासपुर (छ.ग.)

B.C.A. (Third Semester) Examination, Nov.–Dec. 2026
बी.सी.ए. (तृतीय सेमेस्टर) परीक्षा, नवम्बर–दिसम्बर 2026

RELATIONAL DATABASE MANAGEMENT SYSTEM
रिलेशनल डेटाबेस मैनेजमेंट सिस्टम

Time / **समय** : 3 Hours	Maximum Marks / **पूर्णांक** : 70

DEMO SAMPLE – not from the actual sets. *Prepared by Online Student Seva to show the paper pattern and the model-answer style · not an official University question paper*

Instructions:

1. Section A: Q.1 (objective) is compulsory. Attempt any 4 of the 5 short-answer questions in Q.2.
2. Section B: attempt **one** question from each unit (Q.3 to Q.6); each has an internal choice.
3. Figures in the right margin show full marks. Draw neat, labelled diagrams wherever necessary.

निर्देश : खण्ड-अ में प्रश्न 1 अनिवार्य है; प्रश्न 2 के 5 में से 4 प्रश्न हल करें। खण्ड-ब में प्रत्येक इकाई से **एक** प्रश्न हल करें (प्रश्न 3 से 6), सभी में आंतरिक विकल्प है। दाईं ओर के अंक पूर्णांक दर्शाते हैं। आवश्यकतानुसार स्वच्छ व नामांकित चित्र बनाएँ।

SECTION – A / खण्ड – अ

Q.1 **Choose the correct answer.** सही उत्तर चुनिए। 10 × 1 = 10

(1) Which of the following is a function of the DBMS itself, and not of an individual application program?
निम्नलिखित में से कौन-सा कार्य स्वयं DBMS का है, किसी अलग अनुप्रयोग प्रोग्राम का नहीं?

(a) Enforcing integrity constraints on stored data / संग्रहीत डेटा पर अखंडता प्रतिबंधों को लागू करना
(b) Choosing colours for a data-entry screen / डेटा-प्रविष्टि स्क्रीन के रंग चुनना
(c) Printing the company name on a report / रिपोर्ट पर कंपनी का नाम छापना
(d) Validating a mobile number in a web form / वेब फ़ॉर्म में मोबाइल नंबर की जाँच करना

(2) Logical data independence means the ability to change the -
तार्किक डेटा स्वतंत्रता का अर्थ किसे बदल पाने की क्षमता है -

(a) Conceptual schema without rewriting external views and application programs / संप्रत्ययात्मक स्कीमा, बिना बाह्य दृश्यों और अनुप्रयोग प्रोग्रामों को दोबारा लिखे
(b) Disk on which the files are stored / वह डिस्क जिस पर फ़ाइलें संग्रहीत हैं
(c) Indexing method used on a file / फ़ाइल पर प्रयुक्त अनुक्रमण (इंडेक्स) विधि
(d) Operating system of the server / सर्वर का ऑपरेटिंग सिस्टम

(3) In the network data model, a record type can have -
नेटवर्क डेटा मॉडल में एक रिकॉर्ड प्रकार के -

(a) Exactly one parent only / केवल ठीक एक जनक हो सकता है
(b) More than one parent / एक से अधिक जनक हो सकते हैं
(c) No relationship with any other record type / किसी अन्य रिकॉर्ड प्रकार से कोई संबंध नहीं होता
(d) Only one child record / केवल एक संतान रिकॉर्ड होता है

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

(4) While converting an E-R diagram into relations, a many-to-many relationship between STUDENT and COURSE is represented by -
E-R आरेख को संबंधों (रिलेशन) में बदलते समय STUDENT और COURSE के बीच बहु-से-बहु संबंध को कैसे दर्शाया जाता है -

(a) A foreign key added in STUDENT only / केवल STUDENT में एक विदेशी कुंजी जोड़कर
(b) A foreign key added in COURSE only / केवल COURSE में एक विदेशी कुंजी जोड़कर
(c) A separate relation holding the keys of both entities / एक अलग संबंध द्वारा, जिसमें दोनों इकाइयों की कुंजियाँ हों
(d) Merging both entities into one relation / दोनों इकाइयों को एक संबंध में मिलाकर

(5) An attribute whose value can be calculated from other attributes, such as Age from Date_of_Birth, is called a -
वह गुण जिसका मान अन्य गुणों से निकाला जा सके, जैसे जन्मतिथि से आयु, कहलाता है -

(a) Composite attribute / संयुक्त गुण
(b) Multivalued attribute / बहुमान गुण
(c) Derived attribute / व्युत्पन्न गुण
(d) Key attribute / कुंजी गुण

(6) For R(A, B, C, D, E) with the dependencies A -> B, BC -> D and D -> E, the closure of {A, C} is -
R(A, B, C, D, E) में निर्भरताओं A -> B, BC -> D और D -> E के लिए {A, C} का क्लोज़र है -

(a) {A, C} / {A, C}
(b) {A, B, C} / {A, B, C}
(c) {A, B, C, D} / {A, B, C, D}
(d) {A, B, C, D, E} / {A, B, C, D, E}

(7) How many candidate keys does R(A, B, C) have if the dependencies are AB -> C and C -> A?
यदि निर्भरताएँ AB -> C और C -> A हों, तो R(A, B, C) में कितनी उम्मीदवार कुंजियाँ हैं?

(a) One: AB / एक: AB
(b) Two: AB and BC / दो: AB और BC
(c) Two: AB and AC / दो: AB और AC
(d) Three: AB, BC and AC / तीन: AB, BC और AC

(8) Which relational algebra operation is best suited to the query 'find the students who have enrolled in ALL the courses offered'?
'उन विद्यार्थियों को ज्ञात कीजिए जिन्होंने प्रस्तावित सभी पाठ्यक्रमों में नामांकन कराया है' - इस प्रश्न के लिए कौन-सा संबंधपरक बीजगणित संक्रिया सर्वाधिक उपयुक्त है?

(a) Union / यूनियन
(b) Selection / चयन (सिलेक्शन)
(c) Division / विभाजन (डिविज़न)
(d) Natural join / प्राकृतिक जोड़ (नेचुरल जॉइन)

(9) A dirty read occurs when a transaction reads data that has been written by -
डर्टी रीड तब होता है जब कोई ट्रांज़ैक्शन ऐसा डेटा पढ़ता है जिसे लिखा गया हो -

(a) A committed transaction / कमिट हो चुके ट्रांज़ैक्शन द्वारा
(b) An uncommitted transaction that may later roll back / ऐसे अकमिट ट्रांज़ैक्शन द्वारा जो बाद में रोलबैक हो सकता है
(c) The database administrator / डेटाबेस प्रशासक द्वारा
(d) The operating system / ऑपरेटिंग सिस्टम द्वारा

(10) If the precedence graph of a schedule has no cycle, the schedule is -
यदि किसी शेड्यूल के पूर्ववर्ती ग्राफ़ (प्रेसिडेंस ग्राफ़) में कोई चक्र न हो, तो शेड्यूल है -

(a) Not serializable / क्रमबद्ध नहीं (नॉट सीरियलाइज़ेबल)
(b) Conflict serializable / संघर्ष क्रमबद्ध (कॉन्फ्लिक्ट सीरियलाइज़ेबल)
(c) Always a serial schedule / सदा एक क्रमिक शेड्यूल
(d) Certain to deadlock / निश्चित रूप से गतिरोधग्रस्त

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

Q.2 **Attempt any 4 of the following 5 questions.** निम्नलिखित 5 में से किन्हीं 4 प्रश्नों के उत्तर दीजिए। **4 × 5 = 20**

(a) Explain any four functions of a DBMS with a banking example. **5**
बैंकिंग उदाहरण सहित DBMS के किन्हीं चार कार्यों को समझाइए।

(b) What are the ON DELETE options of a foreign key? Explain with an example. **5**
विदेशी कुंजी के ON DELETE विकल्प क्या हैं? उदाहरण सहित समझाइए।

(c) A relation STUDENT(Roll, Name, Dept, HOD) has the dependencies Roll -> Name, Roll -> Dept and Dept -> HOD. Find its key and highest normal form, and normalize it. **5**
संबंध STUDENT(Roll, Name, Dept, HOD) में निर्भरताएँ Roll -> Name, Roll -> Dept और Dept -> HOD हैं। इसकी कुंजी और उच्चतम सामान्य रूप ज्ञात कीजिए तथा इसे सामान्यीकृत कीजिए।

(d) Given EMP(empno, ename, job, sal, deptno), write an SQL query to list the departments whose average salary is more than 35,000, with the average. Show the working on sample data. **5**
EMP(empno, ename, job, sal, deptno) दिया है। उन विभागों की सूची औसत वेतन सहित निकालने के लिए SQL क्वेरी लिखिए जिनका औसत वेतन 35,000 से अधिक है। नमूना डेटा पर कार्यविधि दिखाइए।

(e) Test whether the schedule S: r1(X), r2(X), w1(X), w2(X) is conflict serializable. Compare it with S': r1(X), w1(X), r2(X), w2(X). **5**
जाँचिए कि शेड्यूल S: r1(X), r2(X), w1(X), w2(X) संघर्ष क्रमबद्ध है या नहीं। इसकी तुलना S': r1(X), w1(X), r2(X), w2(X) से कीजिए।

SECTION – B / खण्ड – ब

ONLINE STUDENT SEVA · PARIKSHA PASS

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

Long answer questions · one question from each unit with internal choice / **दीर्घ उत्तरीय प्रश्न**

Q.3 Explain the main functions of a DBMS. Show how a DBMS enforces integrity and security, using SQL for a bank database that has constraints and a view, and test the constraints on sample data. **10**

DBMS के प्रमुख कार्यों को समझाइए। एक बैंक डेटाबेस के लिए, जिसमें प्रतिबंध और एक व्यू हो, SQL की सहायता से दिखाइए कि DBMS अखंडता और सुरक्षा कैसे लागू करता है, तथा प्रतिबंधों की नमूना डेटा पर जाँच कीजिए।

OR / अथवा

Consider EMP(empno, ename, job, sal, deptno) and DEPT(deptno, dname, loc). Write SQL queries, with their output on the sample data, for: (a) employees earning more than 30,000 in descending order of salary; (b) the number of employees and the average and maximum salary of each department; (c) employees earning more than the company average; (d) employee and department names for departments not located at Durg; (e) the number of employees in every department, including departments with none; (f) a 10 per cent raise for clerks.

EMP(empno, ename, job, sal, deptno) और DEPT(deptno, dname, loc) पर विचार कीजिए। नमूना डेटा पर आउटपुट सहित SQL क्वेरी लिखिए: (अ) 30,000 से अधिक वेतन पाने वाले कर्मचारी, वेतन के घटते क्रम में; (ब) प्रत्येक विभाग में कर्मचारियों की संख्या तथा औसत और अधिकतम वेतन; (स) कंपनी के औसत से अधिक वेतन पाने वाले कर्मचारी; (द) उन विभागों के कर्मचारी और विभाग के नाम जो दुर्ग में स्थित नहीं हैं; (य) प्रत्येक विभाग में कर्मचारियों की संख्या, उन विभागों सहित जिनमें कोई नहीं है; (र) क्लर्कों को 10 प्रतिशत वेतन वृद्धि।

ONLINE STUDENT SEVA · PARIKSHA PASS

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

Q.4 A hospital stores data about doctors, patients, wards, appointments and bills. A doctor sees many patients and a patient may see many doctors; each appointment has a date and a fee. A patient is admitted to one ward and a ward holds many patients. A bill is identified only together with its patient. Draw the E-R diagram, identify the entities, attributes, keys, cardinalities and participation, and convert the design into relations with SQL CREATE statements. **10**

एक अस्पताल डॉक्टरों, रोगियों, वार्डों, नियुक्तियों और बिलों का डेटा रखता है। एक डॉक्टर अनेक रोगियों को देखता है और एक रोगी अनेक डॉक्टरों को दिखा सकता है; प्रत्येक नियुक्ति की एक तिथि और शुल्क होता है। रोगी एक वार्ड में भर्ती होता है और एक वार्ड में अनेक रोगी रहते हैं। बिल की पहचान केवल अपने रोगी के साथ होती है। E-R आरेख बनाइए, इकाइयाँ, गुण, कुंजियाँ, कार्डिनैलिटी और सहभागिता पहचानिए तथा डिज़ाइन को SQL CREATE कथनों सहित संबंधों में बदलिए।

OR / अथवा

Explain the degree of a relationship set with examples of unary (recursive), binary and ternary relationships. Show with sample data why the ternary relationship SUPPLIES among supplier, part and project cannot always be replaced by three binary relationships, and show how it and a recursive EMPLOYEE-manages relationship are converted into relations.

संबंध समुच्चय की घात (डिग्री) को एकल (पुनरावर्ती), द्विआधारी और त्रिआधारी संबंधों के उदाहरणों सहित समझाइए। नमूना डेटा से दिखाइए कि आपूर्तिकर्ता, पुर्ज़े और परियोजना के बीच त्रिआधारी संबंध SUPPLIES को सदा तीन द्विआधारी संबंधों से नहीं बदला जा सकता, और यह तथा पुनरावर्ती EMPLOYEE-manages संबंध संबंधों में कैसे बदले जाते हैं।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

Q.5 For the relation R(A, B, C, D, E) with the functional dependencies A -> BC, CD -> E, B -> D and E -> A: (a) find all candidate keys using attribute closure; (b) find a minimal cover; (c) decide whether R is in 3NF and in BCNF; (d) decompose R into BCNF and test the decomposition for lossless join and dependency preservation. **10**

संबंध R(A, B, C, D, E) में प्रकार्यात्मक निर्भरताएँ A -> BC, CD -> E, B -> D और E -> A हैं। (अ) गुण क्लोज़र से सभी उम्मीदवार कुंजियाँ ज्ञात कीजिए; (ब) न्यूनतम आवरण (मिनिमल कवर) ज्ञात कीजिए; (स) निर्णय कीजिए कि R 3NF और BCNF में है या नहीं; (द) R को BCNF में विघटित कीजिए और अपघटन की हानिरहित जोड़ और निर्भरता-संरक्षण के लिए जाँच कीजिए।

OR / अथवा

An order table ORDER_LINE(OrderNo, OrderDate, CustID, CustName, CustCity, ItemNo, ItemName, UnitPrice, Qty) stores one row per item ordered, with key (OrderNo, ItemNo). Starting from the unnormalized sample data, normalize it step by step to 1NF, 2NF and 3NF, state the dependencies used and the anomalies removed, and verify that the decomposition is lossless.

ऑर्डर तालिका ORDER_LINE(OrderNo, OrderDate, CustID, CustName, CustCity, ItemNo, ItemName, UnitPrice, Qty) प्रत्येक ऑर्डर किए गए मद के लिए एक पंक्ति संग्रहीत करती है, जिसकी कुंजी (OrderNo, ItemNo) है। अ-सामान्यीकृत नमूना डेटा से आरंभ कर इसे चरणबद्ध रूप से 1NF, 2NF और 3NF तक सामान्यीकृत कीजिए, प्रयुक्त निर्भरताएँ और हटाई गई विसंगतियाँ बताइए, तथा सत्यापित कीजिए कि अपघटन हानिरहित है।

Q.6 Explain multiple granularity locking. Describe the granularity hierarchy, the lock modes IS, IX, S, SIX and X, the compatibility matrix and the locking rules. With an example, show how three transactions lock the hierarchy and which one has to wait. **10**

बहु-कणिकता लॉकिंग (मल्टीपल ग्रैन्युलैरिटी लॉकिंग) समझाइए। कणिकता पदानुक्रम, लॉक मोड IS, IX, S, SIX और X, संगतता आव्यूह और लॉकिंग नियमों का वर्णन कीजिए। उदाहरण से दिखाइए कि तीन ट्रांज़ैक्शन पदानुक्रम को कैसे लॉक करते हैं और किसे प्रतीक्षा करनी पड़ती है।

OR / अथवा

Explain log-based recovery: the write-ahead log, deferred and immediate database modification, and checkpoints. A system crashes after the log given in the answer has been written. Find the transactions to be redone and undone and the final values of the data items.

लॉग-आधारित पुनःप्राप्ति समझाइए: राइट-अहेड लॉग, आस्थगित और तात्कालिक डेटाबेस संशोधन, तथा चेकपॉइंट। उत्तर में दिया गया लॉग लिखे जाने के बाद सिस्टम क्रैश हो जाता है। दोबारा किए जाने (रीडू) और रद्द किए जाने (अनडू) वाले ट्रांज़ैक्शन तथा डेटा मदों के अंतिम मान ज्ञात कीजिए।

— END OF QUESTION PAPER · प्रश्न-पत्र समाप्त —

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

Syllabus Coverage & Reference Books पाठ्यक्रम कवरेज एवं संदर्भ ग्रंथ

This paper follows the official **Relational Database Management System** syllabus (CASC-08T) of the Chhattisgarh Higher Education Department (NEP-2020). The table shows what each unit contains and where this paper tests it.

Unit	Syllabus content (as prescribed)	Questions in this set
1	**Overview of Database Management** Data processing versus data management; data models: network, relational, hierarchical; instance and schema; file-oriented versus database-oriented approach; data independence; DBMS architecture; DBA roles; DDL, DML, DCL, TCL; kinds of DBMS users; data dictionary. **डेटाबेस प्रबंधन का अवलोकन** — डेटा प्रोसेसिंग बनाम डेटा प्रबंधन; डेटा मॉडल: नेटवर्क, रिलेशनल, हायरार्किकल; इंस्टेंस व स्कीमा; फ़ाइल-ओरिएंटेड बनाम डेटाबेस-ओरिएंटेड दृष्टिकोण; डेटा इंडिपेंडेंस; DBMS आर्किटेक्चर; DBA की भूमिकाएँ; DDL, DML, DCL, TCL; DBMS के उपयोगकर्ता; डेटा डिक्शनरी।	Q.1(1), Q.1(2), Q.1(3), Q.2(a), Q.3
2	**Database Design and E-R Model** Entity, strong and weak entities, relationship, cardinality, attributes; keys; ER diagram; constraints; Codd's rules; extended ER features: generalization, specialization, aggregation, participation; converting the E-R model into a relational schema. **डेटाबेस डिज़ाइन एवं ई-आर मॉडल** — एंटिटी, स्ट्रॉन्ग व वीक एंटिटी, रिलेशनशिप, कार्डिनैलिटी, एट्रिब्यूट; की; ई-आर आरेख; बाध्यताएँ; कॉड के नियम; विस्तारित ई-आर: जनरलाइज़ेशन, स्पेशलाइज़ेशन, एग्रीगेशन, पार्टिसिपेशन; ई-आर मॉडल से रिलेशनल स्कीमा में रूपांतरण।	Q.1(4), Q.1(5), Q.2(b), Q.4
3	**Relational Database Design and Operations** Functional, multivalued and join dependencies; anomalies; decomposition; normal forms 1NF, 2NF, 3NF, BCNF, 4NF, 5NF; de-normalization; relational algebra: select, project, union, Cartesian product, intersection, join; inner, outer and self joins. **रिलेशनल डेटाबेस डिज़ाइन एवं संक्रियाएँ** — फंक्शनल, मल्टीवैल्यूड व जॉइन डिपेंडेंसी; एनॉमली; डिकम्पोज़िशन; नॉर्मल फॉर्म 1NF, 2NF, 3NF, BCNF, 4NF, 5NF; डी-नॉर्मलाइज़ेशन; रिलेशनल अलजेब्रा: सेलेक्ट, प्रोजेक्ट, यूनियन, कार्टेशियन प्रोडक्ट, इंटरसेक्शन, जॉइन; इनर, आउटर व सेल्फ जॉइन।	Q.1(6), Q.1(7), Q.1(8), Q.2(c), Q.2(d), Q.5
4	**Transaction** Transaction; ACID properties; concurrency control techniques; serializability. **ट्रांज़ैक्शन** — ट्रांज़ैक्शन; ACID गुण; समवर्ती नियंत्रण (कंकरेंसी कंट्रोल) तकनीकें; सीरियलाइज़ेबिलिटी।	Q.1(9), Q.1(10), Q.2(e), Q.6

Books recommended in the syllabus (derive your answers from these) / पाठ्यक्रम में अनुशंसित पुस्तकें

Text books:

- Abraham Silberschatz, Henry F. Korth & S. Sudarshan - Database System Concepts (7th Edition); McGraw-Hill Education
- C.J. Date - An Introduction to Database Systems (8th Edition); Pearson / Narosa Publishing House
- Alexis Leon & Mathews Leon - Database Management Systems; Vikas Publishing House, New Delhi

Reference books:

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

- Ramez Elmasri & Shamkant B. Navathe - Fundamentals of Database Systems (7th Edition); Pearson Education
- Jeffrey D. Ullman - Principles of Database Systems; Galgotia / Computer Science Press
- Ivan Bayross - SQL, PL/SQL: The Programming Language of Oracle (4th Edition); BPB Publications, New Delhi

Always use the latest edition. Model answers in this paper follow the standard university textbooks listed above, in simplified exam-ready form. / नवीनतम संस्करण का प्रयोग करें।

About this Demo Set / इस डेमो सेट के बारे में

This is a **DEMO SET**. It follows the exact pattern of the university paper (same number of questions, marks and internal choices) so that you can see how our model answers are written. Its questions are practice questions chosen for demonstration; they are **not taken from the actual Guessing Paper sets**. Do not depend on any single paper – prepare the complete syllabus from your text book.

यह एक **डेमो सेट** है। यह विश्वविद्यालय के प्रश्न-पत्र के ठीक उसी प्रारूप (प्रश्नों की संख्या, अंक और आंतरिक विकल्प) में बना है, ताकि आप हमारे आदर्श उत्तरों की शैली देख सकें। इसके प्रश्न प्रदर्शन के लिए चुने गए अभ्यास-प्रश्न हैं; ये **असली गेसिंग पेपर सेटों में से नहीं लिए गए हैं**। किसी एक पेपर पर निर्भर न रहें - पूरा पाठ्यक्रम अपनी पाठ्यपुस्तक से तैयार करें।

ONLINE STUDENT SEVA · PARIKSHA PASS

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

PART – I : English Medium – Model Answers

Every answer is written to the marks allotted — learn the structure, then write in your own words.

Section A · Q.1 : Answer key

No.	Ans.	Correct option	Explanation
1	(a)	Enforcing integrity constraints on stored data	A DBMS centrally enforces integrity, security, concurrency control and recovery for all programs. Screen layout, report headings and form checks belong to the application.
2	(a)	Conceptual schema without rewriting external views and application programs	Logical data independence protects the external level from changes in the conceptual level, such as adding a new attribute or splitting a table. It is harder to achieve than physical independence.
3	(b)	More than one parent	The network model is a graph in which a child record may be linked to several parents, which allows many-to-many structures. The hierarchical model allows only one parent.
4	(c)	A separate relation holding the keys of both entities	An M:N relationship needs its own relation (for example ENROLMENT) whose primary key is the combination of the two foreign keys, plus any attributes of the relationship.
5	(c)	Derived attribute	A derived attribute is not stored but computed when needed, and it is shown by a dashed ellipse in an E-R diagram.
6	(d)	{A, B, C, D, E}	Start with {A, C}. A -> B adds B, then BC -> D adds D, and D -> E adds E, so the closure is {A, B, C, D, E}. Since it contains every attribute, AC is a candidate key.
7	(b)	Two: AB and BC	AB+ = {A, B, C} and BC+ = {A, B, C} (C gives A), so both are keys. AC does not determine B, and B alone is not enough, so there are exactly two keys.
8	(c)	Division	The division operator R / S returns the values of one attribute that are paired in R with every tuple of S, which is the meaning of 'for all' in a query.
9	(b)	An uncommitted transaction that may later roll back	Reading a value that another transaction has written but not yet committed is a dirty read. If that transaction rolls back, the reader has used a value that never officially existed.
10	(b)	Conflict serializable	A schedule is conflict serializable if and only if its precedence graph is acyclic, and a topological order of the graph gives an equivalent serial schedule.

Section A · Q.2 : Short answers

Q.2(a) 5 marks · ≈ 150–170 words

Explain any four functions of a DBMS with a banking example.

A DBMS manages data on behalf of all users and programs. Its main functions are:

- **Data definition and storage:** it stores tables such as ACCOUNT and CUSTOMER and keeps their descriptions in the data dictionary.

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

- **Data manipulation:** users insert, update, delete and query records using a query language such as SQL.
- **Integrity and security:** it rejects invalid data, for example a negative balance, and allows only authorised users to see or change data.
- **Concurrency and recovery:** many tellers can update accounts at the same time without errors, and after a crash the database is restored to its last consistent state.

In a bank, these functions keep every account correct, private and available all day.

Q.2(b) 5 marks · ≈ 150–170 words

What are the ON DELETE options of a foreign key? Explain with an example.

A foreign key must always match an existing primary key. The ON DELETE option decides what happens to the child rows when the parent row is deleted.

- **CASCADE:** the child rows are deleted automatically.
- **SET NULL:** the foreign key in the child rows is set to NULL.
- **RESTRICT (or NO ACTION):** the deletion of the parent is refused while child rows exist.

Example: in EMP(empno, deptno) with deptno referring to DEPT, deleting department 10 under CASCADE removes its employees; under SET NULL the employees stay with deptno = NULL; under RESTRICT the delete fails with an error. The choice depends on whether the child data is meaningful without its parent.

Q.2(c) 5 marks · ≈ 150–170 words

A relation STUDENT(Roll, Name, Dept, HOD) has the dependencies Roll -> Name, Roll -> Dept and Dept -> HOD. Find its key and highest normal form, and normalize it.

Roll+ = {Roll, Name, Dept, HOD}, so **Roll is the only candidate key**.

All attributes are atomic, so the relation is in 1NF. The key is a single attribute, so no partial dependency exists and it is in 2NF. But Roll -> Dept and Dept -> HOD give the transitive dependency Roll -> HOD, where HOD is a non-prime attribute depending on a non-key attribute Dept, so it is **not in 3NF**. Highest normal form = **2NF**.

Decomposition into 3NF:

- STUDENT(**Roll**, Name, Dept)
- DEPARTMENT(**Dept**, HOD)

The common attribute Dept is the key of DEPARTMENT, so the join is lossless and both dependencies are preserved.

Q.2(d) 5 marks · ≈ 150–170 words

Given EMP(empno, ename, job, sal, deptno), write an SQL query to list the departments whose

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

average salary is more than 35,000, with the average. Show the working on sample data.

SELECT deptno, AVG(sal) FROM emp GROUP BY deptno HAVING AVG(sal) > 35000;

WHERE filters individual rows before grouping, whereas HAVING filters the groups after aggregation, so a condition on AVG must go in HAVING. GROUP BY deptno forms one group per department, and the aggregate functions COUNT, SUM, MIN and MAX are used in the same way.

Sample data: department 10 has salaries 24,000 and 52,000; department 20 has 30,000, 28,000 and 56,000; department 30 has 22,000 and 45,000.

deptno	Salaries	AVG(sal)	AVG > 35,000?
10	24,000; 52,000	38,000	Yes
20	30,000; 28,000; 56,000	38,000	Yes
30	22,000; 45,000	33,500	No

Result: departments 10 and 20, each with average 38,000.

Q.2(e) 5 marks · ≈ 150–170 words

Test whether the schedule S: r1(X), r2(X), w1(X), w2(X) is conflict serializable. Compare it with S': r1(X), w1(X), r2(X), w2(X).

Two operations conflict if they belong to different transactions, use the same item and at least one is a write. An edge Ti -> Tj is drawn when a conflicting operation of Ti comes before that of Tj.

Schedule S: r2(X) comes before w1(X), giving T2 -> T1. Also r1(X) comes before w2(X), giving T1 -> T2. The graph has a **cycle**, so S is **not conflict serializable**; it is the lost-update pattern, because both read the old X.

Schedule S': every conflicting pair has T1 first (w1(X) before r2(X) and w2(X)), so there is only T1 -> T2. No cycle, so S' is conflict serializable and equivalent to the serial order T1, T2.

Section B : Long answers

Q.3 · Unit 1 10 marks · ≈ 600–700 words

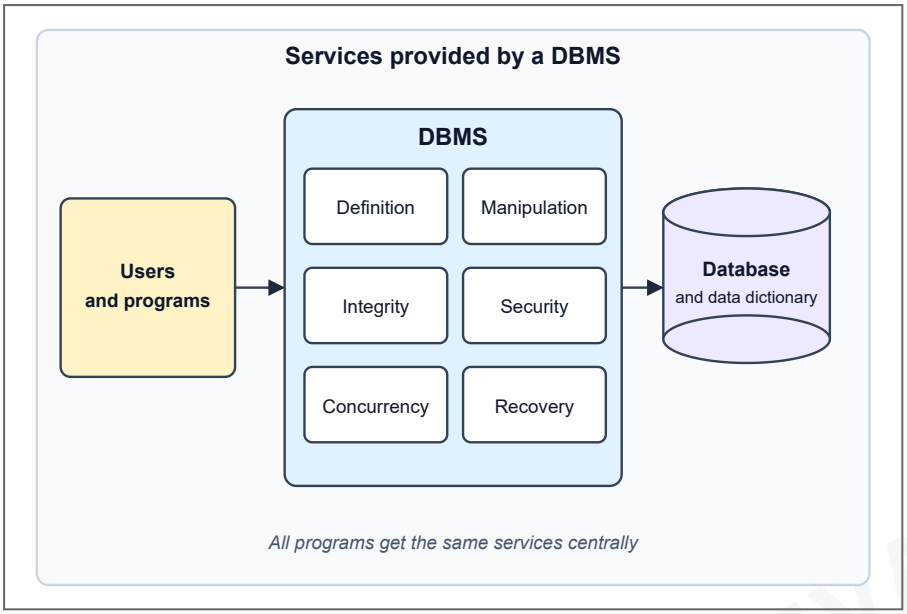
Explain the main functions of a DBMS. Show how a DBMS enforces integrity and security, using SQL for a bank database that has constraints and a view, and test the constraints on sample data.

INTRODUCTION

A DBMS is software that creates, stores, protects and retrieves data for many users and programs at the same time. In a file system every program handles its own files and repeats the same checks. A DBMS instead provides a fixed set of services centrally, so that all programs get the same correct and safe data. This answer explains those functions and then shows, with SQL for a small bank database, how integrity and security are enforced.

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

A. MAIN FUNCTIONS OF A DBMS



Services provided by a DBMS between the users and the stored data

- **Data definition.** Tables, columns, data types and constraints are declared with DDL, and their description is kept in the **data dictionary**.
- **Storage and retrieval.** The DBMS decides how records are placed on disk, builds indexes and fetches the required rows quickly.
- **Data manipulation.** Users insert, update, delete and query data with DML or SQL without knowing the physical layout.
- **Integrity enforcement.** Rules such as unique keys, valid ranges and valid references are checked automatically on every change.
- **Security and authorization.** Only identified users receive only the rights given to them.
- **Concurrency control.** Many transactions run together without spoiling each other's results.
- **Backup and recovery.** After a crash or power failure the database is returned to its last consistent state.
- **Data independence.** Storage or schema changes do not force every program to be rewritten.

B. ENFORCING INTEGRITY WITH CONSTRAINTS

Consider a bank with two tables.

CREATE TABLE branch (br_code CHAR(3) PRIMARY KEY, br_name VARCHAR(30) NOT NULL UNIQUE);

CREATE TABLE account (acc_no INT PRIMARY KEY, holder VARCHAR(30) NOT NULL, balance INT CHECK (balance >= 0), br_code CHAR(3) REFERENCES branch(br_code));

Sample data: branch ('RPR', 'Raipur Main') and account (1001, 'Asha', 5000, 'RPR'). The following insertions were tested.

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

Statement	Result	Constraint that acted
Insert account 1001 again for 'Bina'	Rejected	PRIMARY KEY (entity integrity)
Insert account 1002 with balance -50	Rejected	CHECK (domain integrity)
Insert account 1003 with holder NULL	Rejected	NOT NULL
Insert account 1004 with br_code 'XXX'	Rejected	FOREIGN KEY (referential integrity)
Insert account 1002 for 'Bina', balance 800, branch 'RPR'	Accepted	All rules satisfied

Because these rules are stored in the database, no application program can bypass them, and the database never holds a duplicate account number, a negative balance or an account of a non-existent branch.

C. ENFORCING SECURITY WITH VIEWS AND PRIVILEGES

A **view** is a stored query that shows only selected columns. A clerk who handles address and branch work need not see balances.

CREATE VIEW acc_public AS SELECT acc_no, holder, br_code FROM account;

After this, the table itself can stay hidden and only the view is exposed.

GRANT SELECT ON acc_public TO clerk;

GRANT SELECT, UPDATE ON account TO teller;

REVOKE UPDATE ON account FROM teller;

Here the clerk can read account number, holder and branch but can never see the balance column, the teller can read and change accounts until the update right is revoked, and everyone else has no access. Real systems add user authentication with passwords, encryption of stored and transmitted data, and an audit trail of who changed what.

D. WHY CENTRAL ENFORCEMENT IS BETTER

- A rule is written once in the database, not copied into every program, so it cannot be forgotten in one of them.
- A new rule, such as a minimum balance of 500, is added in one place with an ALTER statement.
- The DBA can inspect the data dictionary to see who has which privilege.
- Violations are caught at the moment of the change, not found later in reports.

CONCLUSION

A DBMS acts as the single guardian of shared data. It defines, stores, manipulates, protects and recovers data, and through constraints, views and privileges it keeps the bank's data correct and confidential. This is the main reason a DBMS is preferred to separate files.

Q.3 (OR) · Unit 1 10 marks · ≈ 600–700 words

Consider EMP(empno, ename, job, sal, deptno) and DEPT(deptno, dname, loc). Write SQL queries, with their output on the sample data, for: (a) employees earning more than 30,000 in descending order of salary; (b) the number of employees and the average and maximum salary of each department; (c) employees earning more than the company average; (d) employee and

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

department names for departments not located at Durg; (e) the number of employees in every department, including departments with none; (f) a 10 per cent raise for clerks.

INTRODUCTION

SQL is the standard language of a relational DBMS. Its data manipulation part has four statements: SELECT to read, INSERT to add, UPDATE to change and DELETE to remove rows. A SELECT query is built from clauses written in the order SELECT, FROM, WHERE, GROUP BY, HAVING and ORDER BY. All answers below were run on the following sample data.

DEPT

deptno	dname	loc
10	Accounts	Raipur
20	Sales	Bilaspur
30	Stores	Durg
40	Research	Raigarh

EMP

empno	ename	job	sal	deptno
101	Asha	Clerk	24,000	10
102	Bhaskar	Manager	52,000	10
103	Chitra	Salesman	30,000	20
104	Deepak	Salesman	28,000	20
105	Esha	Manager	56,000	20
106	Farhan	Clerk	22,000	30
107	Gauri	Analyst	45,000	30

(A) SALARY ABOVE 30,000, HIGHEST FIRST

SELECT ename, sal FROM emp WHERE sal > 30000 ORDER BY sal DESC;

ename	sal
Esha	56,000
Bhaskar	52,000
Gauri	45,000

WHERE filters rows, and Chitra, whose salary is exactly 30,000, is correctly left out because the condition is strictly greater.

(B) DEPARTMENT-WISE COUNT, AVERAGE AND MAXIMUM

SELECT deptno, COUNT(empno), AVG(sal), MAX(sal) FROM emp GROUP BY deptno;

deptno	COUNT	AVG(sal)	MAX(sal)
10	2	38,000	52,000
20	3	38,000	56,000
30	2	33,500	45,000

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

GROUP BY makes one group per department, and the aggregate functions work on each group. Department 40 does not appear because it has no employee row.

(C) ABOVE THE COMPANY AVERAGE (SUBQUERY)

SELECT ename FROM emp WHERE sal > (SELECT AVG(sal) FROM emp);

The inner query finds the company average, 2,57,000 / 7 = 36,714.29. The outer query returns **Bhaskar, Esha and Gauri**, whose salaries are above it. The inner query is evaluated first, and the answer changes automatically when the data change.

(D) JOIN WITH A CONDITION

SELECT e.ename, d.dname FROM emp e JOIN dept d ON e.deptno = d.deptno WHERE d.loc <> 'Durg' ORDER BY e.ename;

ename	dname
Asha	Accounts
Bhaskar	Accounts
Chitra	Sales
Deepak	Sales
Esha	Sales

The join condition matches each employee with the department row having the same deptno. Farhan and Gauri are dropped because their department is located at Durg.

(E) COUNT INCLUDING EMPTY DEPARTMENTS (OUTER JOIN)

SELECT d.dname, COUNT(e.empno) FROM dept d LEFT JOIN emp e ON d.deptno = e.deptno GROUP BY d.dname ORDER BY d.dname;

dname	COUNT
Accounts	2
Research	0
Sales	3
Stores	2

A LEFT JOIN keeps every department even if no employee matches, and COUNT of a column ignores the NULLs, so Research shows 0. An ordinary inner join would have lost the Research row.

(F) UPDATE

UPDATE emp SET sal = sal + sal / 10 WHERE job = 'Clerk';

Asha's salary becomes 26,400 and Farhan's becomes 24,200. Without the WHERE clause, every employee would have been changed, so the condition must be written carefully.

INSERT AND DELETE

INSERT INTO emp VALUES (108, 'Hemant', 'Clerk', 21000, 40);

DELETE FROM emp WHERE empno = 108;

The first statement adds one employee to the Research department, which would then show a count of 1 in part (e). The second removes only the row that satisfies the condition. A DELETE without a WHERE clause removes every row of the table.

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

ORDER OF PROCESSING AND COMMON MISTAKES

A query is written with SELECT first, but it is processed as FROM and joins, WHERE, GROUP BY, HAVING, SELECT and finally ORDER BY. This is why WHERE cannot contain an aggregate such as AVG, which belongs in HAVING.

- Writing = NULL instead of IS NULL gives no rows, because NULL is never equal to anything.
- Forgetting the join condition gives a Cartesian product of 7 x 4 = 28 rows.
- Using WHERE instead of HAVING for a condition on a group gives an error.

CONCLUSION

These six queries show the main SQL tools: filtering with WHERE, sorting, grouping with aggregates, subqueries, joins, outer joins and UPDATE. Most practical queries are combinations of these few clauses.

Q.4 · Unit 2 10 marks · ≈ 600–700 words

A hospital stores data about doctors, patients, wards, appointments and bills. A doctor sees many patients and a patient may see many doctors; each appointment has a date and a fee. A patient is admitted to one ward and a ward holds many patients. A bill is identified only together with its patient. Draw the E-R diagram, identify the entities, attributes, keys, cardinalities and participation, and convert the design into relations with SQL CREATE statements.

INTRODUCTION

E-R design turns a description of a real situation into entities, attributes and relationships, and the diagram is then converted into relations that a relational DBMS can store. The hospital case has five entity types. Each step is shown below, and the final SQL was tested on sample data.

A. ENTITIES, ATTRIBUTES AND KEYS

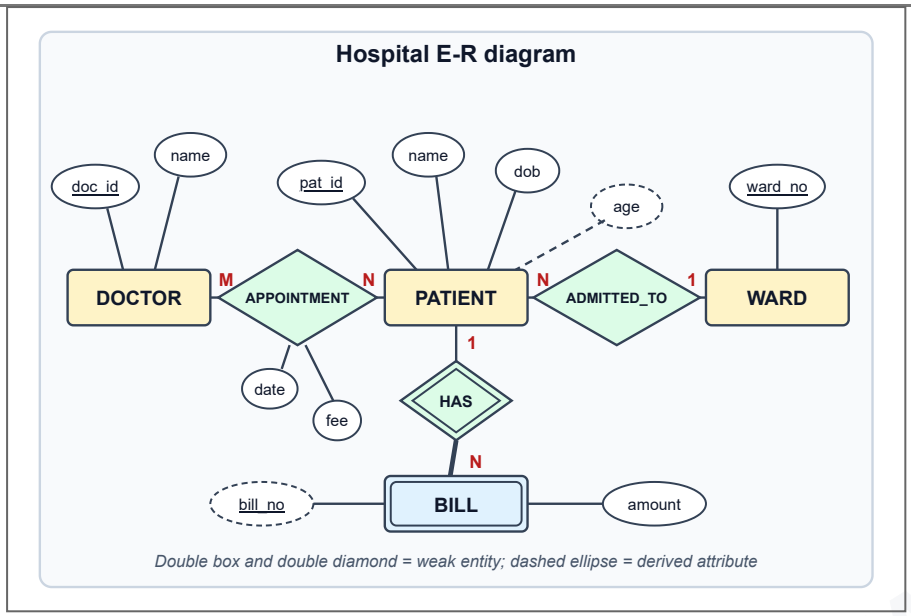
Entity	Attributes	Key	Type
DOCTOR	doc_id, name, speciality	doc_id	Strong
PATIENT	pat_id, name, dob, phone, age	pat_id	Strong
WARD	ward_no, ward_type, beds	ward_no	Strong
BILL	bill_no, amount	bill_no (partial key)	Weak

Age is a **derived attribute**, calculated from dob, so it is shown with a dashed ellipse and is not stored. If a patient could have several phone numbers, phone would be a multivalued attribute and would need a separate table.

B. RELATIONSHIPS, CARDINALITY AND PARTICIPATION

Relationship	Between	Cardinality	Participation
APPOINTMENT (date, fee)	DOCTOR - PATIENT	M : N	Partial for both
ADMITTED_TO	PATIENT - WARD	N : 1	Partial for patient, partial for ward
HAS (identifying)	PATIENT - BILL	1 : N	Total for BILL

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।



E-R diagram of the hospital database

BILL is a **weak entity** because a bill number such as 1 is repeated for every patient. It is identified only by the patient's key together with its partial key bill_no, and so its participation in HAS is total.

C. RULES USED FOR CONVERSION

1. Each strong entity becomes a relation with its simple attributes and primary key.
2. An M : N relationship becomes a separate relation holding both keys and the relationship's own attributes.
3. An N : 1 relationship is stored by adding the key of the one side as a foreign key on the many side.
4. A weak entity becomes a relation whose key is the owner's key plus the partial key, with a foreign key to the owner.
5. Derived attributes are dropped.

D. RELATIONAL SCHEMA IN SQL

CREATE TABLE ward (ward_no INT PRIMARY KEY, ward_type VARCHAR(15), beds INT);

CREATE TABLE doctor (doc_id INT PRIMARY KEY, name VARCHAR(30) NOT NULL, speciality VARCHAR(20));

CREATE TABLE patient (pat_id INT PRIMARY KEY, name VARCHAR(30) NOT NULL, dob DATE, phone CHAR(10), ward_no INT REFERENCES ward(ward_no));

CREATE TABLE appointment (doc_id INT REFERENCES doctor(doc_id), pat_id INT REFERENCES patient(pat_id) ON DELETE CASCADE, appt_date DATE, fee INT, PRIMARY KEY (doc_id, pat_id, appt_date));

CREATE TABLE bill (pat_id INT REFERENCES patient(pat_id) ON DELETE CASCADE, bill_no INT, amount INT, PRIMARY KEY (pat_id, bill_no));

Including appt_date in the key of APPOINTMENT lets the same doctor see the same patient on different days.

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

E. TESTING THE DESIGN

Sample data: wards 1 (General) and 2 (ICU); doctors Rao and Sen; patients Meena (ward 1) and Karan (ward 2); three appointments; three bills, of which two belong to Meena.

- Inserting a patient with ward_no 9 is **rejected**, because ward 9 does not exist (referential integrity).
- Deleting Meena removes her appointments (3 rows become 1) and her bills (3 rows become 1) through ON DELETE CASCADE, so no bill is left without an owner.
- Two patients can each have a bill numbered 1 without conflict, because the key is (pat_id, bill_no).

F. A QUERY ON THE SCHEMA

To find the total billed amount of each patient:

SELECT p.name, SUM(b.amount) FROM patient p JOIN bill b ON p.pat_id = b.pat_id GROUP BY p.name;

On the sample data this gives Meena 2,400 (900 + 1,500) and Karan 4,000. Joining PATIENT, APPOINTMENT and DOCTOR in the same way shows that Meena has seen both Dr Rao and Dr Sen, which is the many-to-many relationship at work.

CONCLUSION

The design has four tables for entities, one of them weak, and one relation for the M : N appointment, while the N : 1 admission is a foreign key in PATIENT. Keys and foreign keys enforce the cardinalities, and the SQL schema was tested to behave as the E-R model intends.

Q.4 (OR) · Unit 2 10 marks · ≈ 600–700 words

Explain the degree of a relationship set with examples of unary (recursive), binary and ternary relationships. Show with sample data why the ternary relationship SUPPLIES among supplier, part and project cannot always be replaced by three binary relationships, and show how it and a recursive EMPLOYEE-manages relationship are converted into relations.

INTRODUCTION

The **degree** of a relationship set is the number of entity sets that take part in it. Most relationships in practice have degree two, but some real situations involve one entity set related to itself, or three entity sets related together. A designer who forces such cases into binary form can lose information, as the sample data below show.

A. DEGREES OF RELATIONSHIP

Degree	Name	Example
1	Unary (recursive)	EMPLOYEE manages EMPLOYEE
2	Binary	STUDENT enrols in COURSE
3	Ternary	SUPPLIER supplies PART to PROJECT
n	n-ary	Rare; usually reduced to simpler forms

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

B. UNARY (RECURSIVE) RELATIONSHIP

In EMPLOYEE manages EMPLOYEE, the same entity set plays two **roles**, manager and subordinate, and the roles are written on the lines of the diagram. One manager supervises many subordinates, so the cardinality is 1 : N. It is converted by a foreign key that refers to the same table.

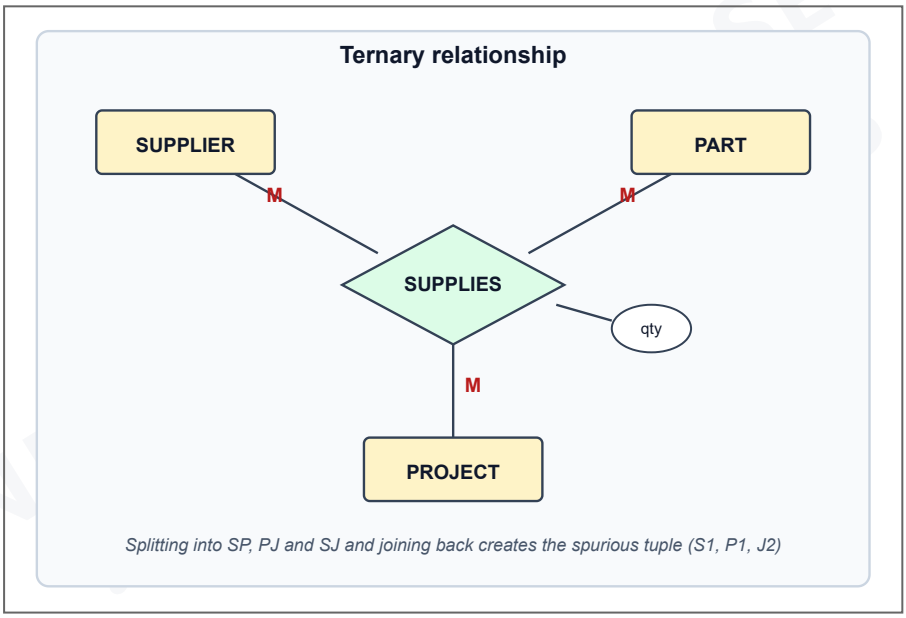
CREATE TABLE employee (emp_id INT PRIMARY KEY, name VARCHAR(30), manager_id INT REFERENCES employee(emp_id));

Sample rows: (1, Rao, NULL), (2, Sen, 1), (3, Das, 1), (4, Eka, 2). Rao has no manager, so participation in the subordinate role is partial. A self join lists each employee with the manager.

SELECT w.name, m.name FROM employee w JOIN employee m ON w.manager_id = m.emp_id;

Subordinate	Manager
Sen	Rao
Das	Rao
Eka	Sen

C. TERNARY RELATIONSHIP



Ternary relationship SUPPLIES among SUPPLIER, PART and PROJECT

A supplier supplies a part to a project with a quantity. The relationship is about the three together, and the quantity belongs to the combination, so it is a ternary relationship with the attribute qty. It is converted into one relation holding the three keys.

CREATE TABLE supplies (sno CHAR(2) REFERENCES supplier, pno CHAR(2) REFERENCES part, jno CHAR(2) REFERENCES project, qty INT, PRIMARY KEY (sno, pno, jno));

D. WHY THREE BINARY RELATIONSHIPS ARE NOT ENOUGH

Suppose SUPPLIES contains only three facts: (S1, P1, J1), (S1, P2, J2) and (S2, P1, J2). Break it into three binary relations by projection.

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

SP (supplier, part)	PJ (part, project)	SJ (supplier, project)
S1, P1	P1, J1	S1, J1
S1, P2	P2, J2	S1, J2
S2, P1	P1, J2	S2, J2

Now join them back. The tuple **(S1, P1, J2)** satisfies all three binary tables, because S1 supplies P1, P1 goes to J2 and S1 works for J2, yet it was never recorded. The join returns four tuples instead of three, so one **spurious tuple** appears and the original facts cannot be recovered. A ternary relationship is therefore not equal to its binary parts unless a special constraint, a join dependency, holds.

E. PARTICIPATION AND KEY CHOICE

All three entity sets take part with partial participation. The key of SUPPLIES is normally all three keys together. If a rule says that each (part, project) pair has only one supplier, then the key reduces to (pno, jno) and sno becomes an ordinary attribute.

F. OTHER RECURSIVE EXAMPLES

A recursive relationship need not be 1 : N. PERSON is married to PERSON is 1 : 1 and can be stored as a spouse_id column in the same table. COURSE is a prerequisite of COURSE is M : N, because a course can have several prerequisites and can itself be a prerequisite of several courses, so it needs a separate relation.

CREATE TABLE prerequisite (course_id CHAR(5), prereq_id CHAR(5), PRIMARY KEY (course_id, prereq_id));

Both columns are foreign keys to COURSE, and the roles course and prerequisite distinguish them. The same rule applies as for any M : N relationship.

CONCLUSION

Degree tells how many entity sets a relationship connects. Unary relationships need role names and a self-referencing foreign key, binary relationships are mapped by keys or by a separate table, and ternary relationships must stay as one relation with three foreign keys, because splitting them can create false facts.

Q.5 · Unit 3 10 marks · ≈ 600–700 words

For the relation R(A, B, C, D, E) with the functional dependencies A -> BC, CD -> E, B -> D and E -> A: (a) find all candidate keys using attribute closure; (b) find a minimal cover; (c) decide whether R is in 3NF and in BCNF; (d) decompose R into BCNF and test the decomposition for lossless join and dependency preservation.

INTRODUCTION

Questions of this kind are solved in a fixed order: find the keys from attribute closures, simplify the dependencies to a minimal cover, test the normal forms, and decompose if a violation is found. The given dependencies are A -> BC, CD -> E, B -> D and E -> A. Every step below was also checked by a short program.

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

(A) CANDIDATE KEYS

The closure X+ is the set of attributes that X determines. Every attribute of R appears on the right side of some dependency, so no attribute is compulsory in every key, and the single attributes and pairs are tested.

X	Closure X+	Key?
A	A, B, C, D, E (A gives B, C; B gives D; CD gives E)	Yes
E	A, B, C, D, E (E gives A, then as above)	Yes
B	B, D	No
C	C	No
D	D	No
BC	A, B, C, D, E (B gives D; CD gives E; E gives A)	Yes, minimal
CD	A, B, C, D, E (CD gives E; E gives A; A gives B)	Yes, minimal

BC and CD are minimal because B, C and D alone do not determine R. The candidate keys are therefore **A, E, BC and CD**, and all five attributes are **prime**.

(B) MINIMAL COVER

1. **Single attribute on the right:** A -> B, A -> C, CD -> E, B -> D, E -> A.
2. **Extraneous attributes on the left:** in CD -> E, C+ = {C} and D+ = {D}, neither contains E, so neither C nor D can be dropped.
3. **Redundant dependencies:** removing a dependency and recomputing the closure of its left side never gives its right side back (for example without B -> D the closure of B is just {B}, and without A -> B the closure of A is {A, C}), so none is redundant.

The minimal cover is {A -> B, A -> C, CD -> E, B -> D, E -> A}, the same five dependencies.

(C) NORMAL FORMS

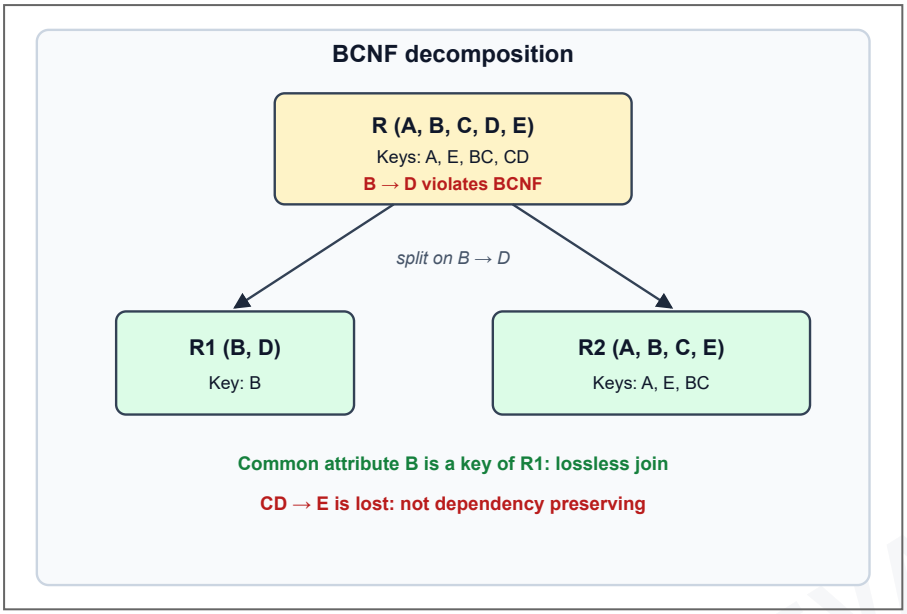
A dependency X -> Y is acceptable in **3NF** if X is a superkey or Y is a prime attribute, and in **BCNF** only if X is a superkey.

Dependency	X a superkey?	Y prime?	3NF	BCNF
A -> BC	Yes	Yes	OK	OK
CD -> E	Yes	Yes	OK	OK
B -> D	No	Yes (D)	OK	**Violated**
E -> A	Yes	Yes	OK	OK

So R is in **3NF but not in BCNF**, because of B -> D. The redundancy is visible in data: two rows (a1, b1, c1, d1, e1) and (a2, b1, c2, d1, e2) store d1 twice, because the same b1 always determines d1.

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

(D) DECOMPOSITION INTO BCNF



BCNF decomposition of R using the violating dependency B -> D

Split on B -> D: **R1(B, D)** and **R2(A, B, C, E)**, which is R without D.

- R1 has the key B and the only dependency B -> D, so it is in BCNF.
- In R2 the keys are A, E and BC, and every determinant (A, E, BC) is a superkey, so R2 is in BCNF.

Lossless join: R1 and R2 share only B, and B -> D makes B a key of R1, so the decomposition is lossless.

Dependency preservation: A -> BC and E -> A hold inside R2, and B -> D holds inside R1. But **CD -> E** needs C and D together, and C lies only in R2 while D lies only in R1. The closure of CD using the projected dependencies is just {C, D}, so this dependency is lost and can be checked only after a join.

CHOOSING BETWEEN THE DESIGNS

R was already in 3NF with all dependencies preserved. BCNF removes the repeated d values but loses CD -> E. A designer who values easy enforcement of every dependency keeps the 3NF design, while one who values freedom from redundancy accepts BCNF and enforces CD -> E through a check or trigger.

CONCLUSION

R has four candidate keys (A, E, BC, CD), a minimal cover of five dependencies, is in 3NF but not BCNF, and its BCNF decomposition into R1(B, D) and R2(A, B, C, E) is lossless but not dependency preserving.

Q.5 (OR) · Unit 3 10 marks · ≈ 600–700 words

An order table ORDER_LINE(OrderNo, OrderDate, CustID, CustName, CustCity, ItemNo, ItemName, UnitPrice, Qty) stores one row per item ordered, with key (OrderNo, ItemNo).

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

Starting from the unnormalized sample data, normalize it step by step to 1NF, 2NF and 3NF, state the dependencies used and the anomalies removed, and verify that the decomposition is lossless.

INTRODUCTION

Normalization removes redundancy by splitting a table according to its functional dependencies. The dependencies of this order table are:

- OrderNo -> OrderDate, CustID
- CustID -> CustName, CustCity
- ItemNo -> ItemName, UnitPrice
- (OrderNo, ItemNo) -> Qty

STEP 0: UNNORMALIZED DATA

Order 501 (5 Oct, customer C1 Meena, Raipur) has items I1 Pen at 10 (quantity 5) and I2 Notebook at 40 (quantity 3). Order 502 (6 Oct, C2 Karan, Durg) has I1 Pen (quantity 10). Order 503 (8 Oct, C1 Meena, Raipur) has I3 Folder at 25 (quantity 4) and I2 Notebook (quantity 2). Each order is one record holding a **repeating group** of items, which violates 1NF.

STEP 1: FIRST NORMAL FORM

Every cell must hold one atomic value, so each item gets its own row, and the key becomes (OrderNo, ItemNo).

OrderNo	OrderDate	CustID	CustName	CustCity	ItemNo	ItemName	UnitPrice	Qty
501	05-Oct	C1	Meena	Raipur	I1	Pen	10	5
501	05-Oct	C1	Meena	Raipur	I2	Notebook	40	3
502	06-Oct	C2	Karan	Durg	I1	Pen	10	10
503	08-Oct	C1	Meena	Raipur	I3	Folder	25	4
503	08-Oct	C1	Meena	Raipur	I2	Notebook	40	2

Anomalies: Meena's name and city are stored in 4 of the 5 rows. An update anomaly arises if her city is changed in only some rows. A new item such as a stapler cannot be inserted until someone orders it, because the key needs an OrderNo. Deleting order 502 also destroys the only record of customer Karan, a deletion anomaly.

STEP 2: SECOND NORMAL FORM

Partial dependencies exist because OrderNo and ItemNo alone determine non-key attributes. The table is split so that each non-key attribute depends on the whole key of its table.

- ORDERS(**OrderNo**, OrderDate, CustID, CustName, CustCity): 3 rows
- ITEM(**ItemNo**, ItemName, UnitPrice): 3 rows
- ORDER_LINE(**OrderNo, ItemNo**, Qty): 5 rows

Item details are now stored once, so Pen and its price appear in one row only.

STEP 3: THIRD NORMAL FORM

In ORDERS there is a **transitive dependency** OrderNo -> CustID -> CustName, CustCity. Customer facts are moved to their own table.

- CUSTOMER holds (C1, Meena, Raipur) and (C2, Karan, Durg).

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

- ORDERS holds (501, 05-Oct, C1), (502, 06-Oct, C2) and (503, 08-Oct, C1).

The final design has four tables: **CUSTOMER(CustID, CustName, CustCity)**, **ORDERS(OrderNo, OrderDate, CustID)**, **ITEM(ItemNo, ItemName, UnitPrice)** and **ORDER_LINE(OrderNo, ItemNo, Qty)**. In each table every determinant is a key, so the design is also in BCNF.

WHY STOP ONLY AT 3NF, NOT AT 2NF

After Step 2 the table ORDERS still stores Meena's name and city for both order 501 and order 503. If she moves to Bilaspur, two rows must be changed, and a missed row leaves the database inconsistent. Only Step 3 stores each customer once, so the update is made in one place. Normalization is complete when every non-key attribute depends on the key, the whole key and nothing but the key.

VERIFICATION OF LOSSLESS JOIN

Joining ORDER_LINE with ORDERS on OrderNo, ORDERS with CUSTOMER on CustID, and ORDER_LINE with ITEM on ItemNo was run on the sample data. It returns exactly the **5 original rows** with no extra or missing row, because each join column is a key of one of the joined tables.

ANOMALIES AFTER NORMALIZATION

- **Update:** Meena's city is changed in one row of CUSTOMER.
- **Insertion:** the stapler is added to ITEM without any order.
- **Deletion:** deleting order 502 removes the ORDERS and ORDER_LINE rows, but Karan stays in CUSTOMER.

The original table held 45 cells and the four tables hold 39, and more importantly each fact is stored once.

CONCLUSION

By removing the repeating group, then partial dependencies, then transitive dependencies, the single table was split into CUSTOMER, ORDERS, ITEM and ORDER_LINE. The split is lossless and removes the update, insertion and deletion anomalies.

Q.6 · Unit 4 10 marks · ≈ 600–700 words

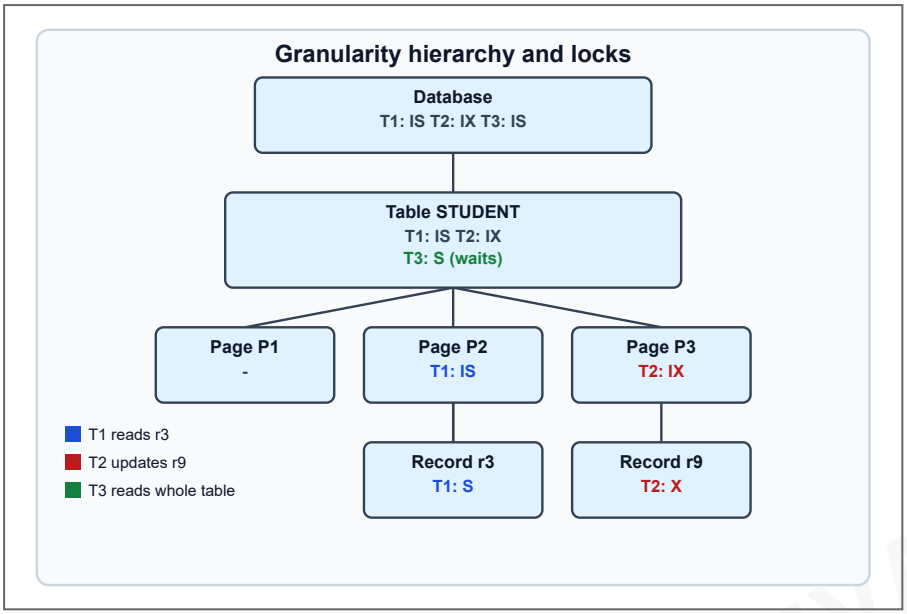
Explain multiple granularity locking. Describe the granularity hierarchy, the lock modes IS, IX, S, SIX and X, the compatibility matrix and the locking rules. With an example, show how three transactions lock the hierarchy and which one has to wait.

INTRODUCTION

Ordinary locking uses one fixed size of data item. If the item is a whole table, a transaction that updates a single row blocks everyone else. If the item is a single record, a transaction that scans a million records must obtain a million locks. **Multiple granularity locking** removes this dilemma by allowing locks at several levels, so that a large reader locks the table once and a small updater locks only a record.

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

A. GRANULARITY HIERARCHY



Granularity hierarchy and the locks taken by three transactions

Data items are arranged as a tree: the **database** contains **tables** (files), a table contains **pages**, and a page contains **records**. Locking a node locks all its descendants implicitly. A transaction that locks a table in S mode therefore reads every page and record below it without any further lock.

B. LOCK MODES

- **S (shared):** the node and its descendants may be read.
- **X (exclusive):** the node and its descendants may be read and written.
- **IS (intention shared):** some descendant will be locked in S mode.
- **IX (intention exclusive):** some descendant will be locked in X or S mode.
- **SIX (shared and intention exclusive):** the whole subtree is read in S mode and some descendants will be locked in X mode.

Intention locks are placed on the ancestors. They warn a transaction that wants a coarse lock that someone is already working lower down, without examining every descendant.

C. COMPATIBILITY MATRIX

Held / Requested	IS	IX	S	SIX	X
IS	Yes	Yes	Yes	Yes	No
IX	Yes	Yes	No	No	No
S	Yes	No	Yes	No	No
SIX	Yes	No	No	No	No
X	No	No	No	No	No

D. LOCKING RULES

1. A lock must obey the compatibility matrix.
2. Locking starts at the root of the tree.
3. To get S or IS on a node, the transaction must hold IS or IX on its parent.
4. To get X, IX or SIX on a node, it must hold IX or SIX on its parent.

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

5. Locks follow two-phase locking, and they are released from the leaf towards the root.

E. EXAMPLE

Take the table STUDENT with pages P1, P2 and P3.

Transaction	Operation	Locks requested in order
T1	Read record r3 in page P2	IS on database, IS on STUDENT, IS on P2, S on r3
T2	Update record r9 in page P3	IX on database, IX on STUDENT, IX on P3, X on r9
T3	Read the whole table STUDENT	IS on database, then S on STUDENT

- **T1 and T2 run together.** At the database and STUDENT, IS and IX are compatible, and they use different pages and records, so neither waits.
- **T3 must wait.** It can take IS on the database, but S on STUDENT conflicts with the IX held by T2, because the matrix gives No for S against IX. T3 proceeds only after T2 releases its locks, and it would have no conflict with T1, since S and IS are compatible.

If T3 had wanted to read the table and update a few rows, it would ask for SIX on STUDENT, then X on those records. SIX allows other transactions to hold only IS, so they can still read individual records.

F. ADVANTAGES AND LIMITATIONS

- Small transactions get high concurrency, while large ones need only a few locks, which lowers the overhead of the lock table.
- The DBMS can choose the level automatically, for example by moving from record locks to a table lock after many records have been locked (lock escalation).
- The protocol is more complex, and deadlocks can still occur, so deadlock detection or prevention is still required.

CONCLUSION

Multiple granularity locking arranges data as a tree and uses intention locks on ancestors so that transactions of different sizes can share the database safely. In the example T1 and T2 run in parallel and the table reader T3 waits for T2.

Q.6 (OR) · Unit 4 10 marks · ≈ 600–700 words

Explain log-based recovery: the write-ahead log, deferred and immediate database modification, and checkpoints. A system crashes after the log given in the answer has been written. Find the transactions to be redone and undone and the final values of the data items.

INTRODUCTION

A crash can occur in the middle of a transaction, leaving the database with some updates written and others lost. The **atomicity** and **durability** parts of ACID are provided by recovery, and the most common method keeps a **log**, a sequential file on stable storage that records every update before it is applied to the database.

A. THE LOG AND THE WRITE-AHEAD RULE

A log record of an update has the form (transaction, data item, old value, new value). Other records mark the start, commit or abort of a transaction. The **write-ahead logging (WAL) rule** says

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

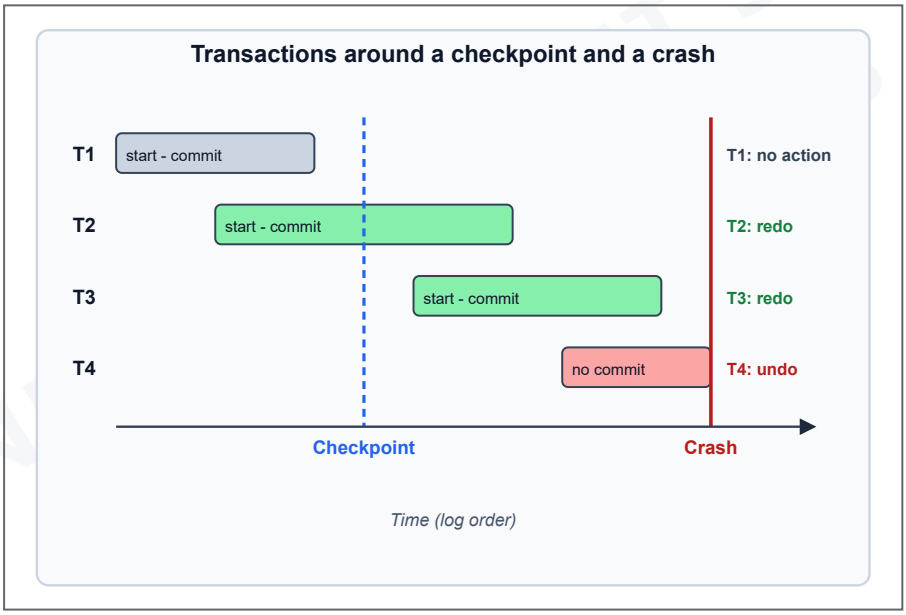
that the log record of an update must reach stable storage before the changed data page is written to the database, and a transaction is called committed only when its commit record has been written to the log. The old value allows an undo and the new value allows a redo.

B. DEFERRED AND IMMEDIATE MODIFICATION

Point	Deferred modification	Immediate modification
When the database is changed	Only after the commit record is logged	Any time during the transaction
Operation needed after a crash	Redo only	Both undo and redo
Old value needed in the log	No	Yes
Uncommitted data on disk	Never	Possible

C. CHECKPOINTS

Without help, recovery would have to read the whole log. At a **checkpoint** the system writes all log records and all modified buffers to disk and then writes a checkpoint record that lists the transactions active at that moment. After a crash only the log from the last checkpoint onwards, and the transactions active at it, need attention.



Timeline of four transactions around a checkpoint and a crash

D. WORKED EXAMPLE

Initial values are A = 1000, B = 2000, C = 700 and D = 500. The log at the time of the crash is:

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

No.	Log record
1	T1 start
2	T1, A, 1000, 900
3	T2 start
4	T2, B, 2000, 2300
5	T1 commit
6	Checkpoint, active transactions: T2
7	T3 start
8	T3, C, 700, 650
9	T2 commit
10	T4 start
11	T4, D, 500, 400
12	T3 commit
-	**Crash**

Building the lists. Start with the undo list holding the transactions active at the checkpoint, here T2, and an empty redo list. Scan forward from the checkpoint. T3 starts and is added to the undo list; T2 commits and moves to the redo list; T4 starts and is added to the undo list; T3 commits and moves to the redo list. The final lists are **redo = T2, T3** and **undo = T4**. T1 committed before the checkpoint, so its update was already forced to disk and it needs no action.

Recovery actions.

1. **Undo T4:** scan the log backward and restore D from 400 to its old value 500, then write an abort record for T4.
2. **Redo T2 and T3:** scan forward from the checkpoint and reapply the new values, so C becomes 650 again if it was lost. T2's update of B was before the checkpoint, so it is already on disk as 2300.

Final values: A = 900, B = 2300, C = 650 and D = 500. Committed transactions T1, T2 and T3 are fully reflected, and the uncommitted transaction T4 has left no trace, which is exactly atomicity and durability.

Effect of the commit record. Had the crash happened just before record 12, T3 would have no commit record, so it would join T4 in the undo list and C would be restored to 700. The commit record alone decides whether a transaction survives, which is why it is forced to disk before the user is told that the transaction is complete.

CONCLUSION

Log-based recovery depends on writing the log first. Deferred modification needs only redo, immediate modification needs undo and redo, and checkpoints limit how much of the log is scanned. In the example T2 and T3 are redone, T4 is undone and the database returns to a consistent state.

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

भाग – II : हिन्दी माध्यम – आदर्श उत्तर

प्रत्येक उत्तर अंक-भार के अनुसार लिखा गया है — ढाँचा सीखें, फिर अपने शब्दों में लिखें।

खण्ड – अ · प्रश्न 1 : उत्तर कुंजी

क्र.	उत्तर	सही विकल्प	स्पष्टीकरण
1	(a)	संग्रहीत डेटा पर अखंडता प्रतिबंधों को लागू करना	DBMS सभी प्रोग्रामों के लिए अखंडता, सुरक्षा, समवर्ती नियंत्रण और पुनःप्राप्ति केंद्रीय रूप से लागू करता है। स्क्रीन का रूप, रिपोर्ट के शीर्षक और फ़ॉर्म की जाँच अनुप्रयोग का काम है।
2	(a)	संप्रत्ययात्मक स्कीमा, बिना बाह्य दृश्यों और अनुप्रयोग प्रोग्रामों को दोबारा लिखे	तार्किक डेटा स्वतंत्रता बाह्य स्तर को संप्रत्ययात्मक स्तर के परिवर्तनों, जैसे नया गुण जोड़ने या तालिका बाँटने, से बचाती है। इसे भौतिक स्वतंत्रता से प्राप्त करना कठिन है।
3	(b)	एक से अधिक जनक हो सकते हैं	नेटवर्क मॉडल ग्राफ़ है जिसमें संतान रिकॉर्ड कई जनकों से जुड़ सकता है, जिससे बहु-से-बहु संरचनाएँ संभव होती हैं। पदानुक्रमित मॉडल में केवल एक जनक होता है।
4	(c)	एक अलग संबंध द्वारा, जिसमें दोनों इकाइयों की कुंजियाँ हों	M:N संबंध के लिए अपना अलग संबंध (जैसे ENROLMENT) चाहिए जिसकी प्राथमिक कुंजी दोनों विदेशी कुंजियों का संयोजन होती है, साथ में संबंध के अपने गुण भी रहते हैं।
5	(c)	व्युत्पन्न गुण	व्युत्पन्न गुण संग्रहीत नहीं किया जाता बल्कि आवश्यकता पड़ने पर निकाला जाता है, और E-R आरेख में उसे बिंदीदार दीर्घवृत्त से दर्शाते हैं।
6	(d)	{A, B, C, D, E}	{A, C} से शुरू करें। A -> B से B जुड़ता है, फिर BC -> D से D और D -> E से E जुड़ता है, अतः क्लोज़र {A, B, C, D, E} है। चूँकि इसमें सभी गुण हैं, AC एक उम्मीदवार कुंजी है।
7	(b)	दो: AB और BC	AB+ = {A, B, C} और BC+ = {A, B, C} (C से A मिलता है), अतः दोनों कुंजियाँ हैं। AC से B निर्धारित नहीं होता और अकेला B पर्याप्त नहीं है, इसलिए ठीक दो कुंजियाँ हैं।
8	(c)	विभाजन (डिविज़न)	विभाजन संक्रिया R / S उस गुण के मान लौटाती है जो R में S के प्रत्येक टपल के साथ जुड़े हों; प्रश्न में 'सभी' का यही अर्थ है।
9	(b)	ऐसे अकमिट ट्रांज़ैक्शन द्वारा जो बाद में रोलबैक हो सकता है	किसी अन्य ट्रांज़ैक्शन द्वारा लिखा पर अभी कमिट न हुआ मान पढ़ना डर्टी रीड है। यदि वह ट्रांज़ैक्शन रोलबैक हो जाए, तो पाठक ने ऐसे मान का प्रयोग किया जो कभी आधिकारिक रूप से था ही नहीं।
10	(b)	संघर्ष क्रमबद्ध (कॉन्फ्लिक्ट सीरियलाइज़ेबल)	शेड्यूल संघर्ष क्रमबद्ध तभी होता है जब उसका पूर्ववर्ती ग्राफ़ चक्रहीन हो, और ग्राफ़ का सांस्थितिक क्रम (टोपोलॉजिकल ऑर्डर) एक समतुल्य क्रमिक शेड्यूल देता है।

खण्ड – अ · प्रश्न 2 : लघु उत्तर

प्रश्न 2(a) — **5 अंक · शब्द-सीमा ≈ 150–170 शब्द**

बैंकिंग उदाहरण सहित DBMS के किन्हीं चार कार्यों को समझाइए।

DBMS सभी उपयोगकर्ताओं और प्रोग्रामों की ओर से डेटा का प्रबंधन करता है। इसके मुख्य कार्य हैं:

- **डेटा परिभाषा और संग्रह:** यह ACCOUNT और CUSTOMER जैसी तालिकाएँ संग्रहीत करता है और उनका विवरण डेटा डिक्शनरी में रखता है।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

- **डेटा मैनिपुलेशन:** उपयोगकर्ता SQL जैसी क्वेरी भाषा से रिकॉर्ड जोड़ते, बदलते, हटाते और खोजते हैं।
- **अखंडता और सुरक्षा:** यह अमान्य डेटा, जैसे ऋणात्मक शेष, को अस्वीकार करता है और केवल अधिकृत उपयोगकर्ताओं को डेटा देखने या बदलने देता है।
- **समवर्ती नियंत्रण और पुनःप्राप्ति:** कई टेलर एक साथ बिना त्रुटि के खाते अद्यतन कर सकते हैं, और क्रैश के बाद डेटाबेस अपनी अंतिम संगत अवस्था में लौटाया जाता है।

बैंक में ये कार्य हर खाते को सही, गोपनीय और दिनभर उपलब्ध रखते हैं।

प्रश्न 2(b) — 5 अंक · शब्द-सीमा ≈ 150–170 शब्द

विदेशी कुंजी के ON DELETE विकल्प क्या हैं? उदाहरण सहित समझाइए।

विदेशी कुंजी सदा किसी विद्यमान प्राथमिक कुंजी से मेल खानी चाहिए। ON DELETE विकल्प तय करता है कि जनक पंक्ति हटने पर संतान पंक्तियों का क्या होगा।

- **CASCADE:** संतान पंक्तियाँ अपने-आप हट जाती हैं।
- **SET NULL:** संतान पंक्तियों में विदेशी कुंजी NULL हो जाती है।
- **RESTRICT (या NO ACTION):** संतान पंक्तियाँ रहने तक जनक का हटाना अस्वीकार किया जाता है।

उदाहरण: EMP(empno, deptno) में deptno, DEPT को संदर्भित करता है। विभाग 10 हटाने पर CASCADE में उसके कर्मचारी हट जाते हैं; SET NULL में कर्मचारी deptno = NULL के साथ बने रहते हैं; RESTRICT में हटाना त्रुटि के साथ विफल होता है। चुनाव इस पर निर्भर है कि संतान डेटा जनक के बिना अर्थपूर्ण है या नहीं।

प्रश्न 2(c) — 5 अंक · शब्द-सीमा ≈ 150–170 शब्द

संबंध STUDENT(Roll, Name, Dept, HOD) में निर्भरताएँ Roll -> Name, Roll -> Dept और Dept -> HOD हैं। इसकी कुंजी और उच्चतम सामान्य रूप ज्ञात कीजिए तथा इसे सामान्यीकृत कीजिए।

Roll+ = {Roll, Name, Dept, HOD}, इसलिए **Roll ही एकमात्र उम्मीदवार कुंजी है**।

सभी गुण परमाणुक हैं, अतः संबंध 1NF में है। कुंजी एकल गुण है, इसलिए आंशिक निर्भरता नहीं है और यह 2NF में है। परंतु Roll -> Dept और Dept -> HOD से संक्रमणीय निर्भरता Roll -> HOD बनती है, जिसमें HOD गैर-मुख्य गुण है और गैर-कुंजी गुण Dept पर निर्भर है, इसलिए यह **3NF में नहीं है**। उच्चतम सामान्य रूप = **2NF**।

3NF में अपघटन:

- STUDENT(**Roll**, Name, Dept)
- DEPARTMENT(**Dept**, HOD)

उभयनिष्ठ गुण Dept, DEPARTMENT की कुंजी है, इसलिए जोड़ हानिरहित है और दोनों निर्भरताएँ संरक्षित रहती हैं।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

प्रश्न 2(d) 5 अंक · शब्द-सीमा ≈ 150–170 शब्द

EMP(empno, ename, job, sal, deptno) दिया है। उन विभागों की सूची औसत वेतन सहित निकालने के लिए SQL क्वेरी लिखिए जिनका औसत वेतन 35,000 से अधिक है। नमूना डेटा पर कार्यविधि दिखाइए।

SELECT deptno, AVG(sal) FROM emp GROUP BY deptno HAVING AVG(sal) > 35000;

WHERE समूह बनाने से पहले अलग-अलग पंक्तियाँ छाँटता है, जबकि HAVING योग-क्रिया के बाद समूहों को छाँटता है, इसलिए AVG पर शर्त HAVING में ही लगेगी। GROUP BY deptno प्रत्येक विभाग का एक समूह बनाता है, और COUNT, SUM, MIN तथा MAX योग-फ़ंक्शन भी इसी प्रकार प्रयुक्त होते हैं।

नमूना डेटा: विभाग 10 में वेतन 24,000 और 52,000; विभाग 20 में 30,000, 28,000 और 56,000; विभाग 30 में 22,000 और 45,000 हैं।

deptno	वेतन	AVG(sal)	AVG > 35,000?
10	24,000; 52,000	38,000	हाँ
20	30,000; 28,000; 56,000	38,000	हाँ
30	22,000; 45,000	33,500	नहीं

परिणाम: विभाग 10 और 20, दोनों का औसत 38,000।

प्रश्न 2(e) 5 अंक · शब्द-सीमा ≈ 150–170 शब्द

जाँचिए कि शेड्यूल S: r1(X), r2(X), w1(X), w2(X) संघर्ष क्रमबद्ध है या नहीं। इसकी तुलना S': r1(X), w1(X), r2(X), w2(X) से कीजिए।

दो संक्रियाएँ टकराती हैं यदि वे अलग ट्रांज़ैक्शनों की हों, एक ही मद पर हों और कम से कम एक लेखन (राइट) हो। जब Ti की टकराने वाली संक्रिया Tj की संक्रिया से पहले आए, तो किनारा Ti -> Tj खींचा जाता है।

शेड्यूल S: r2(X), w1(X) से पहले है, जिससे T2 -> T1 बनता है। साथ ही r1(X), w2(X) से पहले है, जिससे T1 -> T2 बनता है। ग्राफ़ में **चक्र** है, इसलिए S **संघर्ष क्रमबद्ध नहीं** है; यह लॉस्ट-अपडेट का रूप है, क्योंकि दोनों पुराना X पढ़ते हैं।

शेड्यूल S': प्रत्येक टकराने वाले जोड़े में T1 पहले है (w1(X), r2(X) और w2(X) से पहले), इसलिए केवल T1 -> T2 है। चक्र नहीं है, अतः S' संघर्ष क्रमबद्ध है और क्रमिक क्रम T1, T2 के समतुल्य है।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

खण्ड – ब · दीर्घ उत्तर

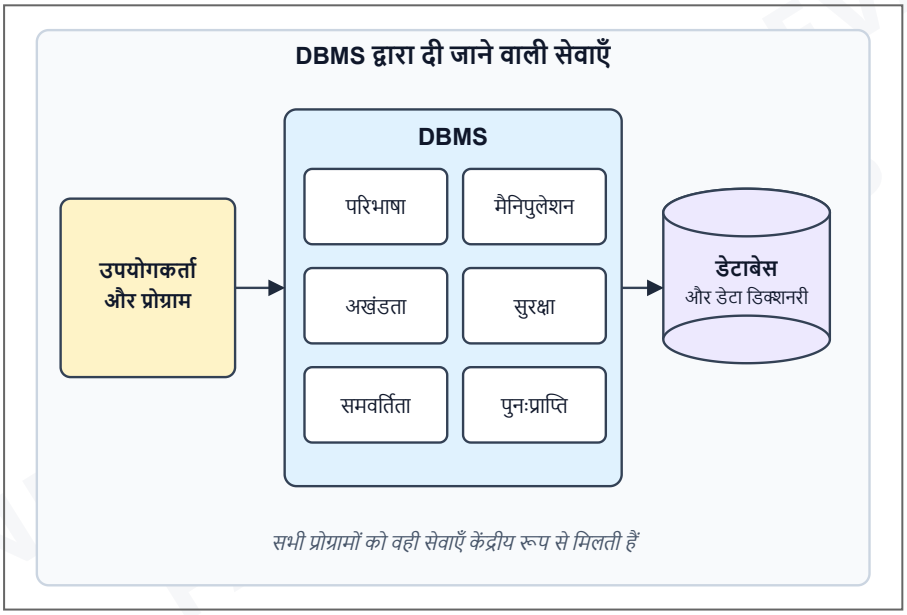
प्रश्न 3 · इकाई 1 10 अंक · ≈ 600–700 शब्द

DBMS के प्रमुख कार्यों को समझाइए। एक बैंक डेटाबेस के लिए, जिसमें प्रतिबंध और एक व्यू हो, SQL की सहायता से दिखाइए कि DBMS अखंडता और सुरक्षा कैसे लागू करता है, तथा प्रतिबंधों की नमूना डेटा पर जाँच कीजिए।

प्रस्तावना

DBMS ऐसा सॉफ़्टवेयर है जो एक ही समय में अनेक उपयोगकर्ताओं और प्रोग्रामों के लिए डेटा बनाता, संग्रहीत करता, सुरक्षित रखता और पुनः प्राप्त करता है। फ़ाइल प्रणाली में प्रत्येक प्रोग्राम अपनी फ़ाइलें स्वयं सँभालता है और वही जाँचें दोहराता है। DBMS इसके बजाय निश्चित सेवाएँ केंद्रीय रूप से देता है, ताकि सभी प्रोग्रामों को वही सही और सुरक्षित डेटा मिले। इस उत्तर में इन कार्यों की व्याख्या है और फिर एक छोटे बैंक डेटाबेस के SQL से दिखाया गया है कि अखंडता और सुरक्षा कैसे लागू होती है।

क. DBMS के प्रमुख कार्य



उपयोगकर्ताओं और संग्रहीत डेटा के बीच DBMS द्वारा दी जाने वाली सेवाएँ

- **डेटा परिभाषा।** DDL से तालिकाएँ, स्तंभ, डेटा प्रकार और प्रतिबंध घोषित किए जाते हैं, और उनका विवरण **डेटा डिक्शनरी** में रखा जाता है।
- **संग्रह और पुनःप्राप्ति।** DBMS तय करता है कि रिकॉर्ड डिस्क पर कहाँ रखे जाएँ, इंडेक्स बनाता है और आवश्यक पंक्तियाँ शीघ्र लाता है।
- **डेटा मैनिपुलेशन।** उपयोगकर्ता भौतिक संरचना जाने बिना DML या SQL से डेटा जोड़ते, बदलते, हटाते और खोजते हैं।
- **अखंडता प्रवर्तन।** अद्वितीय कुंजी, वैध सीमा और वैध संदर्भ जैसे नियम हर परिवर्तन पर अपने-आप जाँचे जाते हैं।
- **सुरक्षा और प्राधिकरण।** केवल पहचाने गए उपयोगकर्ताओं को वही अधिकार मिलते हैं जो उन्हें दिए गए हैं।
- **समवर्ती नियंत्रण।** अनेक ट्रांज़ैक्शन एक-दूसरे के परिणाम बिगाड़े बिना साथ चलते हैं।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

- **बैकअप और पुनःप्राप्ति।** क्रैश या बिजली जाने के बाद डेटाबेस अपनी अंतिम संगत अवस्था में लौटाया जाता है।
- **डेटा स्वतंत्रता।** संग्रह या स्कीमा बदलने पर हर प्रोग्राम दोबारा नहीं लिखना पड़ता।

ख. प्रतिबंधों से अखंडता का प्रवर्तन

दो तालिकाओं वाले एक बैंक पर विचार कीजिए।

CREATE TABLE branch (br_code CHAR(3) PRIMARY KEY, br_name VARCHAR(30) NOT NULL UNIQUE);

CREATE TABLE account (acc_no INT PRIMARY KEY, holder VARCHAR(30) NOT NULL, balance INT CHECK (balance >= 0), br_code CHAR(3) REFERENCES branch(br_code));

नमूना डेटा: branch ('RPR', 'Raipur Main') और account (1001, 'Asha', 5000, 'RPR')। निम्नलिखित प्रविष्टियाँ परखी गईं।

कथन	परिणाम	कार्य करने वाला प्रतिबंध
'Bina' के लिए खाता 1001 फिर जोड़ना	अस्वीकृत	PRIMARY KEY (इकाई अखंडता)
शेष -50 के साथ खाता 1002 जोड़ना	अस्वीकृत	CHECK (डोमेन अखंडता)
धारक NULL के साथ खाता 1003 जोड़ना	अस्वीकृत	NOT NULL
br_code 'XXX' के साथ खाता 1004 जोड़ना	अस्वीकृत	FOREIGN KEY (संदर्भ अखंडता)
'Bina' के लिए खाता 1002, शेष 800, शाखा 'RPR' जोड़ना	स्वीकृत	सभी नियम पूरे

ये नियम डेटाबेस में ही संग्रहीत हैं, इसलिए कोई अनुप्रयोग प्रोग्राम उन्हें लाँघ नहीं सकता, और डेटाबेस में कभी दोहरा खाता नंबर, ऋणात्मक शेष या अस्तित्वहीन शाखा का खाता नहीं रहता।

ग. व्यू और विशेषाधिकारों से सुरक्षा का प्रवर्तन

व्यू संग्रहीत क्वेरी है जो केवल चुने हुए स्तंभ दिखाता है। जो क्लर्क पते और शाखा का काम देखता है, उसे शेष राशि देखने की आवश्यकता नहीं है।

CREATE VIEW acc_public AS SELECT acc_no, holder, br_code FROM account;

इसके बाद मूल तालिका छिपी रह सकती है और केवल व्यू खुला रखा जाता है।

GRANT SELECT ON acc_public TO clerk;

GRANT SELECT, UPDATE ON account TO teller;

REVOKE UPDATE ON account FROM teller;

यहाँ क्लर्क खाता संख्या, धारक और शाखा पढ़ सकता है पर शेष स्तंभ कभी नहीं देख सकता, टेलर अपडेट अधिकार वापस लिए जाने तक खाते पढ़ और बदल सकता है, और अन्य किसी को पहुँच नहीं है। वास्तविक प्रणालियाँ पासवर्ड से उपयोगकर्ता प्रमाणीकरण, संग्रहीत और प्रेषित डेटा का एन्क्रिप्शन, और किसने क्या बदला इसका ऑडिट ट्रेल भी जोड़ती हैं।

घ. केंद्रीय प्रवर्तन बेहतर क्यों है

- नियम डेटाबेस में एक बार लिखा जाता है, हर प्रोग्राम में नहीं दोहराया जाता, इसलिए किसी प्रोग्राम में उसके छूट जाने का खतरा नहीं रहता।
- नया नियम, जैसे 500 का न्यूनतम शेष, ALTER कथन से एक ही स्थान पर जोड़ा जाता है।
- DBA डेटा डिक्शनरी देखकर जान सकता है कि किसके पास कौन-सा विशेषाधिकार है।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

- उल्लंघन परिवर्तन के क्षण में ही पकड़े जाते हैं, बाद में रिपोर्टों में नहीं।

निष्कर्ष

DBMS साझा डेटा के एकमात्र संरक्षक का काम करता है। वह डेटा को परिभाषित, संग्रहीत, संशोधित, सुरक्षित और पुनःप्राप्त करता है, और प्रतिबंधों, व्यू तथा विशेषाधिकारों से बैंक के डेटा को सही और गोपनीय रखता है। अलग-अलग फ़ाइलों की तुलना में DBMS को पसंद किए जाने का यही मुख्य कारण है।

प्रश्न 3 (अथवा) • इकाई 1 10 अंक · ≈ 600–700 शब्द

EMP(empno, ename, job, sal, deptno) और DEPT(deptno, dname, loc) पर विचार कीजिए। नमूना डेटा पर आउटपुट सहित SQL क्वेरी लिखिए: (अ) 30,000 से अधिक वेतन पाने वाले कर्मचारी, वेतन के घटते क्रम में; (ब) प्रत्येक विभाग में कर्मचारियों की संख्या तथा औसत और अधिकतम वेतन; (स) कंपनी के औसत से अधिक वेतन पाने वाले कर्मचारी; (द) उन विभागों के कर्मचारी और विभाग के नाम जो दुर्ग में स्थित नहीं हैं; (य) प्रत्येक विभाग में कर्मचारियों की संख्या, उन विभागों सहित जिनमें कोई नहीं है; (र) क्लर्कों को 10 प्रतिशत वेतन वृद्धि।

प्रस्तावना

SQL संबंधपरक DBMS की मानक भाषा है। इसके डेटा मैनिपुलेशन भाग में चार कथन हैं: पंक्तियाँ पढ़ने के लिए SELECT, जोड़ने के लिए INSERT, बदलने के लिए UPDATE और हटाने के लिए DELETE। SELECT क्वेरी खंडों से बनती है, जो SELECT, FROM, WHERE, GROUP BY, HAVING और ORDER BY के क्रम में लिखे जाते हैं। नीचे सभी उत्तर इस नमूना डेटा पर चलाए गए हैं।

DEPT

deptno	dname	loc
10	Accounts	Raipur
20	Sales	Bilaspur
30	Stores	Durg
40	Research	Raigarh

EMP

empno	ename	job	sal	deptno
101	Asha	Clerk	24,000	10
102	Bhaskar	Manager	52,000	10
103	Chitra	Salesman	30,000	20
104	Deepak	Salesman	28,000	20
105	Esha	Manager	56,000	20
106	Farhan	Clerk	22,000	30
107	Gauri	Analyst	45,000	30

(अ) 30,000 से अधिक वेतन, सर्वाधिक पहले

SELECT ename, sal FROM emp WHERE sal > 30000 ORDER BY sal DESC;

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

ename	sal
Esha	56,000
Bhaskar	52,000
Gauri	45,000

WHERE पंक्तियाँ छाँटता है, और ठीक 30,000 वेतन वाली Chitra सही ढंग से छूट जाती है क्योंकि शर्त कठोर रूप से अधिक की है।

(ब) विभागवार संख्या, औसत और अधिकतम

SELECT deptno, COUNT(empno), AVG(sal), MAX(sal) FROM emp GROUP BY deptno;

deptno	COUNT	AVG(sal)	MAX(sal)
10	2	38,000	52,000
20	3	38,000	56,000
30	2	33,500	45,000

GROUP BY प्रत्येक विभाग का एक समूह बनाता है और योग-फ़ंक्शन प्रत्येक समूह पर काम करते हैं। विभाग 40 नहीं दिखता क्योंकि उसकी कोई कर्मचारी पंक्ति नहीं है।

(स) कंपनी के औसत से अधिक (उप-क्वेरी)

SELECT ename FROM emp WHERE sal > (SELECT AVG(sal) FROM emp);

भीतरी क्वेरी कंपनी का औसत निकालती है, 2,57,000 / 7 = 36,714.29। बाहरी क्वेरी **Bhaskar, Esha और Gauri** लौटाती है, जिनका वेतन इससे अधिक है। भीतरी क्वेरी पहले चलती है, और डेटा बदलने पर उत्तर अपने-आप बदल जाता है।

(द) शर्त सहित जोड़ (जॉइन)

SELECT e.ename, d.dname FROM emp e JOIN dept d ON e.deptno = d.deptno WHERE d.loc <> 'Durg' ORDER BY e.ename;

ename	dname
Asha	Accounts
Bhaskar	Accounts
Chitra	Sales
Deepak	Sales
Esha	Sales

जोड़ की शर्त प्रत्येक कर्मचारी को उसी deptno वाली विभाग पंक्ति से मिलाती है। Farhan और Gauri छूट जाते हैं क्योंकि उनका विभाग दुर्ग में है।

(य) रिक्त विभागों सहित गणना (आउटर जॉइन)

SELECT d.dname, COUNT(e.empno) FROM dept d LEFT JOIN emp e ON d.deptno = e.deptno GROUP BY d.dname ORDER BY d.dname;

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

dname	COUNT
Accounts	2
Research	0
Sales	3
Stores	2

LEFT JOIN हर विभाग को रखता है चाहे कोई कर्मचारी मेल न खाए, और किसी स्तंभ का COUNT NULL को छोड़ देता है, इसलिए Research में 0 दिखता है। सामान्य इनर जॉइन में Research की पंक्ति खो जाती।

(र) अद्यतन (अपडेट)

UPDATE emp SET sal = sal + sal / 10 WHERE job = 'Clerk';

Asha का वेतन 26,400 और Farhan का 24,200 हो जाता है। WHERE खंड के बिना हर कर्मचारी बदल जाता, इसलिए शर्त सावधानी से लिखनी चाहिए।

इन्सर्ट और डिलीट

INSERT INTO emp VALUES (108, 'Hemant', 'Clerk', 21000, 40);

DELETE FROM emp WHERE empno = 108;

पहला कथन Research विभाग में एक कर्मचारी जोड़ता है, जिससे भाग (य) में उसकी गणना 1 दिखेगी। दूसरा केवल वही पंक्ति हटाता है जो शर्त पूरी करती है। WHERE के बिना DELETE तालिका की सभी पंक्तियाँ हटा देता है।

प्रक्रमण का क्रम और सामान्य भूलें

क्वेरी में SELECT पहले लिखा जाता है, पर उसका प्रक्रमण FROM और जोड़, WHERE, GROUP BY, HAVING, SELECT और अंत में ORDER BY के क्रम में होता है। इसीलिए WHERE में AVG जैसा योग-फ़ंक्शन नहीं आ सकता, उसका स्थान HAVING में है।

- = NULL लिखने के बजाय IS NULL लिखना चाहिए; = NULL कोई पंक्ति नहीं देता, क्योंकि NULL किसी के बराबर नहीं होता।
- जोड़ की शर्त भूलने पर 7 x 4 = 28 पंक्तियों का कार्टेशियन गुणनफल मिलता है।
- समूह की शर्त के लिए HAVING के स्थान पर WHERE लगाने पर त्रुटि आती है।

निष्कर्ष

ये छह क्वेरी SQL के मुख्य औज़ार दिखाती हैं: WHERE से छाँटना, क्रम लगाना, योग-फ़ंक्शन सहित समूहन, उप-क्वेरी, जोड़, आउटर जोड़ और UPDATE। अधिकांश व्यावहारिक क्वेरी इन्हीं कुछ खंडों के संयोजन होती हैं।

प्रश्न 4 • इकाई 2 10 अंक · ≈ 600–700 शब्द

एक अस्पताल डॉक्टरों, रोगियों, वार्डों, नियुक्तियों और बिलों का डेटा रखता है। एक डॉक्टर अनेक रोगियों को देखता है और एक रोगी अनेक डॉक्टरों को दिखा सकता है; प्रत्येक नियुक्ति की एक तिथि और शुल्क होता है। रोगी एक वार्ड में भर्ती होता है और एक वार्ड में अनेक रोगी रहते हैं। बिल की

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

पहचान केवल अपने रोगी के साथ होती है। E-R आरेख बनाइए, इकाइयाँ, गुण, कुंजियाँ, कार्डिनैलिटी और सहभागिता पहचानिए तथा डिज़ाइन को SQL CREATE कथनों सहित संबंधों में बदलिए।

प्रस्तावना

E-R डिज़ाइन किसी वास्तविक स्थिति के विवरण को इकाइयों, गुणों और संबंधों में बदलता है, और फिर आरेख को ऐसे संबंधों (रिलेशन) में बदला जाता है जिन्हें संबंधपरक DBMS संग्रहीत कर सके। अस्पताल के मामले में पाँच इकाई प्रकार हैं। नीचे प्रत्येक चरण दिखाया गया है, और अंतिम SQL नमूना डेटा पर परखा गया है।

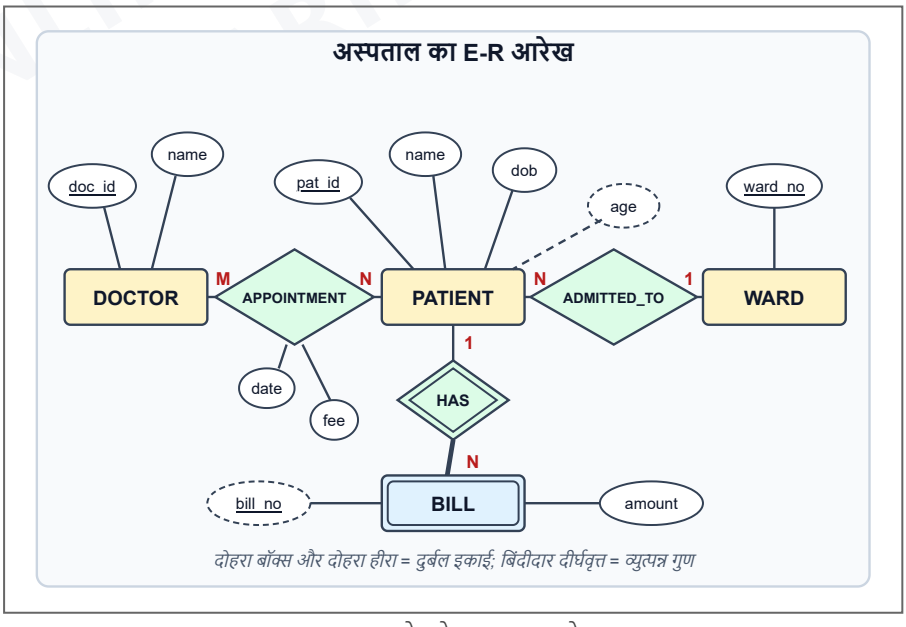
क. इकाइयाँ, गुण और कुंजियाँ

इकाई	गुण	कुंजी	प्रकार
DOCTOR	doc_id, name, speciality	doc_id	प्रबल
PATIENT	pat_id, name, dob, phone, age	pat_id	प्रबल
WARD	ward_no, ward_type, beds	ward_no	प्रबल
BILL	bill_no, amount	bill_no (आंशिक कुंजी)	दुर्बल

आयु **व्युत्पन्न गुण** है, जो dob से निकाली जाती है, इसलिए उसे बिंदीदार दीर्घवृत्त से दर्शाते हैं और संग्रहीत नहीं करते। यदि रोगी के कई फ़ोन नंबर हो सकते, तो phone बहुमान गुण होता और उसके लिए अलग तालिका चाहिए होती।

ख. संबंध, कार्डिनैलिटी और सहभागिता

संबंध	के बीच	कार्डिनैलिटी	सहभागिता
APPOINTMENT (date, fee)	DOCTOR - PATIENT	M : N	दोनों की आंशिक
ADMITTED_TO	PATIENT - WARD	N : 1	रोगी की आंशिक, वार्ड की आंशिक
HAS (पहचान कराने वाला)	PATIENT - BILL	1 : N	BILL की पूर्ण



अस्पताल डेटाबेस का E-R आरेख

BILL **दुर्बल इकाई** है क्योंकि बिल संख्या, जैसे 1, हर रोगी के लिए दोहराई जाती है। उसकी पहचान केवल रोगी की कुंजी और आंशिक कुंजी bill_no के साथ होती है, इसलिए HAS में उसकी सहभागिता पूर्ण है।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

ग. रूपांतरण में प्रयुक्त नियम

1. प्रत्येक प्रबल इकाई अपने सरल गुणों और प्राथमिक कुंजी सहित एक संबंध बनती है।
2. M : N संबंध एक अलग संबंध बनता है जिसमें दोनों कुंजियाँ और संबंध के अपने गुण होते हैं।
3. N : 1 संबंध को एक पक्ष की कुंजी बहु पक्ष में विदेशी कुंजी के रूप में जोड़कर रखा जाता है।
4. दुर्बल इकाई ऐसा संबंध बनती है जिसकी कुंजी स्वामी की कुंजी और आंशिक कुंजी होती है, तथा स्वामी की ओर विदेशी कुंजी रहती है।
5. व्युत्पन्न गुण छोड़ दिए जाते हैं।

घ. SQL में संबंधपरक स्कीमा

CREATE TABLE ward (ward_no INT PRIMARY KEY, ward_type VARCHAR(15), beds INT);

CREATE TABLE doctor (doc_id INT PRIMARY KEY, name VARCHAR(30) NOT NULL, speciality VARCHAR(20));

CREATE TABLE patient (pat_id INT PRIMARY KEY, name VARCHAR(30) NOT NULL, dob DATE, phone CHAR(10), ward_no INT REFERENCES ward(ward_no));

CREATE TABLE appointment (doc_id INT REFERENCES doctor(doc_id), pat_id INT REFERENCES patient(pat_id) ON DELETE CASCADE, appt_date DATE, fee INT, PRIMARY KEY (doc_id, pat_id, appt_date));

CREATE TABLE bill (pat_id INT REFERENCES patient(pat_id) ON DELETE CASCADE, bill_no INT, amount INT, PRIMARY KEY (pat_id, bill_no));

APPOINTMENT की कुंजी में appt_date रखने से एक ही डॉक्टर एक ही रोगी को अलग-अलग दिन देख सकता है।

ङ. डिज़ाइन की जाँच

नमूना डेटा: वार्ड 1 (General) और 2 (ICU); डॉक्टर Rao और Sen; रोगी Meena (वार्ड 1) और Karan (वार्ड 2); तीन नियुक्तियाँ; तीन बिल, जिनमें दो Meena के हैं।

- ward_no 9 वाला रोगी जोड़ना **अस्वीकृत** होता है, क्योंकि वार्ड 9 मौजूद नहीं है (संदर्भ अखंडता)।
- Meena को हटाने पर ON DELETE CASCADE से उसकी नियुक्तियाँ (3 पंक्तियाँ घटकर 1) और बिल (3 पंक्तियाँ घटकर 1) हट जाते हैं, इसलिए कोई बिल बिना स्वामी के नहीं रहता।
- दो रोगियों के पास बिना टकराव के संख्या 1 वाला बिल हो सकता है, क्योंकि कुंजी (pat_id, bill_no) है।

च. स्कीमा पर एक क्वेरी

प्रत्येक रोगी की कुल बिल राशि जानने के लिए:

SELECT p.name, SUM(b.amount) FROM patient p JOIN bill b ON p.pat_id = b.pat_id GROUP BY p.name;

नमूना डेटा पर यह Meena 2,400 (900 + 1,500) और Karan 4,000 देती है। इसी प्रकार PATIENT, APPOINTMENT और DOCTOR को जोड़ने पर दिखता है कि Meena ने Dr Rao और Dr Sen दोनों को दिखाया है, जो बहु-से-बहु संबंध का कार्य है।

निष्कर्ष

डिज़ाइन में इकाइयों के लिए चार तालिकाएँ हैं, जिनमें एक दुर्बल है, और M : N नियुक्ति के लिए एक संबंध है, जबकि N : 1 भर्ती PATIENT में विदेशी कुंजी है। कुंजियाँ और विदेशी कुंजियाँ कार्डिनैलिटी लागू करती हैं, और

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

SQL स्कीमा उसी तरह व्यवहार करता पाया गया जैसा E-R मॉडल चाहता है।

प्रश्न 4 (अथवा) • इकाई 2 10 अंक · ≈ 600–700 शब्द

संबंध समुच्चय की घात (डिग्री) को एकल (पुनरावर्ती), द्विआधारी और त्रिआधारी संबंधों के उदाहरणों सहित समझाइए। नमूना डेटा से दिखाइए कि आपूर्तिकर्ता, पुर्ज़े और परियोजना के बीच त्रिआधारी संबंध SUPPLIES को सदा तीन द्विआधारी संबंधों से नहीं बदला जा सकता, और यह तथा पुनरावर्ती EMPLOYEE-manages संबंध संबंधों में कैसे बदले जाते हैं।

प्रस्तावना

किसी संबंध समुच्चय की **घात (डिग्री)** उसमें भाग लेने वाले इकाई समुच्चयों की संख्या है। व्यवहार में अधिकांश संबंध द्विआधारी (घात दो) होते हैं, पर कुछ वास्तविक स्थितियों में एक इकाई समुच्चय स्वयं से संबंधित होता है, या तीन इकाई समुच्चय एक साथ संबंधित होते हैं। जो डिज़ाइनर ऐसी स्थितियों को ज़बरदस्ती द्विआधारी रूप में ढालता है, वह सूचना खो सकता है, जैसा नीचे नमूना डेटा दिखाता है।

क. संबंध की घातें

घात	नाम	उदाहरण
1	एकल (पुनरावर्ती)	EMPLOYEE manages EMPLOYEE
2	द्विआधारी	STUDENT enrols in COURSE
3	त्रिआधारी	SUPPLIER supplies PART to PROJECT
n	n-आधारी	दुर्लभ; प्रायः सरल रूपों में घटा दिया जाता है

ख. एकल (पुनरावर्ती) संबंध

EMPLOYEE manages EMPLOYEE में एक ही इकाई समुच्चय दो **भूमिकाएँ** निभाता है, प्रबंधक और अधीनस्थ, और आरेख की रेखाओं पर भूमिकाएँ लिखी जाती हैं। एक प्रबंधक अनेक अधीनस्थों का पर्यवेक्षण करता है, इसलिए कार्डिनैलिटी 1 : N है। इसे उसी तालिका की ओर संकेत करने वाली विदेशी कुंजी से बदला जाता है।

CREATE TABLE employee (emp_id INT PRIMARY KEY, name VARCHAR(30), manager_id INT REFERENCES employee(emp_id));

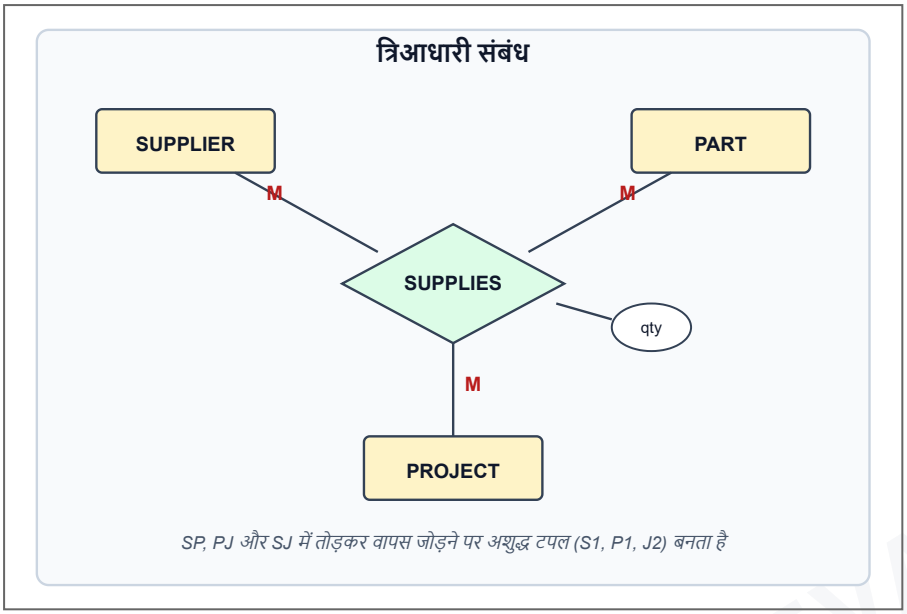
नमूना पंक्तियाँ: (1, Rao, NULL), (2, Sen, 1), (3, Das, 1), (4, Eka, 2)। Rao का कोई प्रबंधक नहीं है, इसलिए अधीनस्थ भूमिका में सहभागिता आंशिक है। सेल्फ जॉइन प्रत्येक कर्मचारी को उसके प्रबंधक के साथ सूचीबद्ध करता है।

SELECT w.name, m.name FROM employee w JOIN employee m ON w.manager_id = m.emp_id;

अधीनस्थ	प्रबंधक
Sen	Rao
Das	Rao
Eka	Sen

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

ग. त्रिआधारी संबंध



SUPPLIER, PART और PROJECT के बीच त्रिआधारी संबंध SUPPLIES

आपूर्तिकर्ता किसी परियोजना को एक पुर्ज़ा मात्रा सहित देता है। संबंध तीनों के बारे में एक साथ है, और मात्रा संयोजन की है, इसलिए यह गुण qty सहित त्रिआधारी संबंध है। इसे तीनों कुंजियों वाले एक संबंध में बदला जाता है।

CREATE TABLE supplies (sno CHAR(2) REFERENCES supplier, pno CHAR(2) REFERENCES part, jno CHAR(2) REFERENCES project, qty INT, PRIMARY KEY (sno, pno, jno));

घ. तीन द्विआधारी संबंध पर्याप्त क्यों नहीं हैं

मान लीजिए SUPPLIES में केवल तीन तथ्य हैं: (S1, P1, J1), (S1, P2, J2) और (S2, P1, J2)। प्रक्षेपण से इसे तीन द्विआधारी संबंधों में तोड़िए।

SP (आपूर्तिकर्ता, पुर्ज़ा)	PJ (पुर्ज़ा, परियोजना)	SJ (आपूर्तिकर्ता, परियोजना)
S1, P1	P1, J1	S1, J1
S1, P2	P2, J2	S1, J2
S2, P1	P1, J2	S2, J2

अब इन्हें वापस जोड़िए। टपल **(S1, P1, J2)** तीनों द्विआधारी तालिकाओं को संतुष्ट करता है, क्योंकि S1, P1 देता है, P1, J2 में जाता है और S1, J2 के लिए काम करता है, फिर भी वह कभी दर्ज नहीं हुआ था। जोड़ तीन के स्थान पर चार टपल लौटाता है, इसलिए एक **अशुद्ध टपल (स्प्यूरियस टपल)** प्रकट होता है और मूल तथ्य वापस नहीं मिल सकते। अतः त्रिआधारी संबंध अपने द्विआधारी भागों के बराबर नहीं होता, जब तक कोई विशेष प्रतिबंध, यानी जॉइन निर्भरता, लागू न हो।

ङ. सहभागिता और कुंजी का चयन

तीनों इकाई समुच्चय आंशिक सहभागिता से भाग लेते हैं। SUPPLIES की कुंजी सामान्यतः तीनों कुंजियाँ मिलाकर होती है। यदि नियम हो कि प्रत्येक (पुर्ज़ा, परियोजना) जोड़ी का एक ही आपूर्तिकर्ता होगा, तो कुंजी घटकर (pno, jno) रह जाती है और sno सामान्य गुण बन जाता है।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

च. अन्य पुनरावर्ती उदाहरण

पुनरावर्ती संबंध का 1 : N होना आवश्यक नहीं। PERSON is married to PERSON 1 : 1 है और उसी तालिका में spouse_id स्तंभ के रूप में रखा जा सकता है। COURSE is a prerequisite of COURSE M : N है, क्योंकि एक पाठ्यक्रम के कई पूर्वापेक्षी हो सकते हैं और वह स्वयं कई पाठ्यक्रमों का पूर्वापेक्षी हो सकता है, इसलिए उसके लिए अलग संबंध चाहिए।

CREATE TABLE prerequisite (course_id CHAR(5), prereq_id CHAR(5), PRIMARY KEY (course_id, prereq_id));

दोनों स्तंभ COURSE की विदेशी कुंजियाँ हैं, और भूमिकाएँ course तथा prerequisite उन्हें अलग करती हैं। वही नियम लागू होता है जो किसी भी M : N संबंध पर होता है।

निष्कर्ष

घात बताती है कि संबंध कितने इकाई समुच्चयों को जोड़ता है। एकल संबंधों को भूमिका नाम और स्व-संदर्भी विदेशी कुंजी चाहिए, द्विआधारी संबंध कुंजियों या अलग तालिका से रूपांतरित होते हैं, और त्रिआधारी संबंध तीन विदेशी कुंजियों वाले एक संबंध के रूप में ही रहने चाहिए, क्योंकि उन्हें तोड़ने से झूठे तथ्य बन सकते हैं।

प्रश्न 5 · इकाई 3 10 अंक · ≈ 600–700 शब्द

संबंध R(A, B, C, D, E) में प्रकार्यात्मक निर्भरताएँ A -> BC, CD -> E, B -> D और E -> A हैं। (अ) गुण क्लोज़र से सभी उम्मीदवार कुंजियाँ ज्ञात कीजिए; (ब) न्यूनतम आवरण (मिनिमल कवर) ज्ञात कीजिए; (स) निर्णय कीजिए कि R 3NF और BCNF में है या नहीं; (द) R को BCNF में विघटित कीजिए और अपघटन की हानिरहित जोड़ और निर्भरता-संरक्षण के लिए जाँच कीजिए।

प्रस्तावना

इस प्रकार के प्रश्न निश्चित क्रम में हल होते हैं: गुण क्लोज़र से कुंजियाँ ज्ञात करना, निर्भरताओं को न्यूनतम आवरण तक सरल करना, सामान्य रूपों की जाँच करना, और उल्लंघन मिलने पर अपघटन करना। दी गई निर्भरताएँ A -> BC, CD -> E, B -> D और E -> A हैं। नीचे हर चरण की जाँच एक छोटे प्रोग्राम से भी की गई है।

(अ) उम्मीदवार कुंजियाँ

क्लोज़र X+ उन गुणों का समुच्चय है जिन्हें X निर्धारित करता है। R का प्रत्येक गुण किसी न किसी निर्भरता के दाएँ पक्ष में आता है, इसलिए कोई गुण हर कुंजी में अनिवार्य नहीं है, और एकल गुणों तथा जोड़ियों की जाँच की जाती है।

X	क्लोज़र X+	कुंजी?
A	A, B, C, D, E (A से B, C; B से D; CD से E)	हाँ
E	A, B, C, D, E (E से A, फिर ऊपर की तरह)	हाँ
B	B, D	नहीं
C	C	नहीं
D	D	नहीं
BC	A, B, C, D, E (B से D; CD से E; E से A)	हाँ, न्यूनतम
CD	A, B, C, D, E (CD से E; E से A; A से B)	हाँ, न्यूनतम

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

BC और CD न्यूनतम हैं क्योंकि B, C और D अकेले R को निर्धारित नहीं करते। अतः उम्मीदवार कुंजियाँ **A, E, BC और CD** हैं, और पाँचों गुण **मुख्य (प्राइम)** हैं।

(ब) न्यूनतम आवरण

1. **दाएँ पक्ष में एक गुण:** A -> B, A -> C, CD -> E, B -> D, E -> A।
2. **बाएँ पक्ष के अनावश्यक गुण:** CD -> E में C+ = {C} और D+ = {D}, किसी में E नहीं है, इसलिए न C हटाया जा सकता है न D।
3. **फ़ालतू निर्भरताएँ:** कोई निर्भरता हटाकर उसके बाएँ पक्ष का क्लोज़र दोबारा निकालने पर दायाँ पक्ष कभी वापस नहीं मिलता (जैसे B -> D के बिना B का क्लोज़र केवल {B} है, और A -> B के बिना A का क्लोज़र {A, C} है), इसलिए कोई फ़ालतू नहीं है।

न्यूनतम आवरण {A -> B, A -> C, CD -> E, B -> D, E -> A} है, यानी वही पाँच निर्भरताएँ।

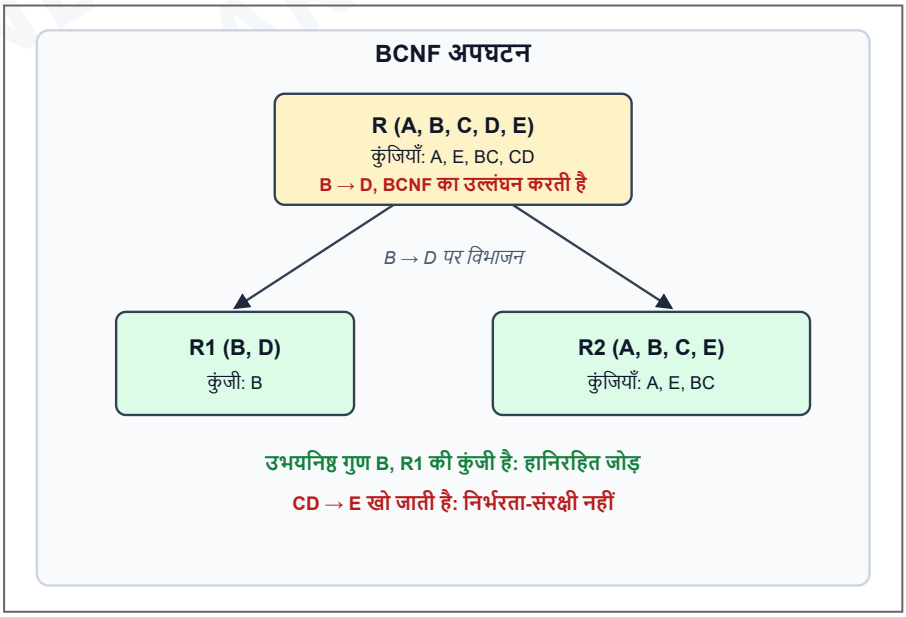
(स) सामान्य रूप

निर्भरता X -> Y **3NF** में स्वीकार्य है यदि X अतिकुंजी हो या Y मुख्य गुण हो, और **BCNF** में तभी जब X अतिकुंजी हो।

निर्भरता	X अतिकुंजी?	Y मुख्य?	3NF	BCNF
A -> BC	हाँ	हाँ	ठीक	ठीक
CD -> E	हाँ	हाँ	ठीक	ठीक
B -> D	नहीं	हाँ (D)	ठीक	**उल्लंघित**
E -> A	हाँ	हाँ	ठीक	ठीक

अतः R **3NF में है पर BCNF में नहीं**, B -> D के कारण। डेटा में पुनरावृत्ति दिखती है: दो पंक्तियाँ (a1, b1, c1, d1, e1) और (a2, b1, c2, d1, e2) d1 को दो बार संग्रहीत करती हैं, क्योंकि वही b1 सदा d1 को निर्धारित करता है।

(द) BCNF में अपघटन



उल्लंघन करने वाली निर्भरता B -> D से R का BCNF अपघटन

B -> D पर विभाजन कीजिए: **R1(B, D)** और **R2(A, B, C, E)**, जो D के बिना R है।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

- R1 की कुंजी B है और एकमात्र निर्भरता B -> D है, इसलिए यह BCNF में है।
- R2 में कुंजियाँ A, E और BC हैं, और हर निर्धारक (A, E, BC) अतिकुंजी है, इसलिए R2 BCNF में है।

हानिरहित जोड़: R1 और R2 में केवल B उभयनिष्ठ है, और B -> D से B, R1 की कुंजी है, इसलिए अपघटन हानिरहित है।

निर्भरता संरक्षण: A -> BC और E -> A, R2 के भीतर तथा B -> D, R1 के भीतर लागू हैं। परंतु **CD -> E** के लिए C और D एक साथ चाहिए, जबकि C केवल R2 में और D केवल R1 में है। प्रक्षेपित निर्भरताओं से CD का क्लोज़र केवल {C, D} है, इसलिए यह निर्भरता खो जाती है और केवल जोड़ के बाद जाँची जा सकती है।

दोनों डिज़ाइनों में चुनाव

R पहले से 3NF में था और सभी निर्भरताएँ संरक्षित थीं। BCNF दोहराए गए d मान हटाता है पर CD -> E खो देता है। जो डिज़ाइनर हर निर्भरता के आसान प्रवर्तन को महत्व देता है वह 3NF डिज़ाइन रखता है, और जो पुनरावृत्ति-मुक्ति को महत्व देता है वह BCNF स्वीकार कर CD -> E को जाँच या ट्रिगर से लागू करता है।

निष्कर्ष

R की चार उम्मीदवार कुंजियाँ (A, E, BC, CD) हैं, न्यूनतम आवरण में पाँच निर्भरताएँ हैं, यह 3NF में है पर BCNF में नहीं, और R1(B, D) तथा R2(A, B, C, E) में इसका BCNF अपघटन हानिरहित है पर निर्भरता-संरक्षी नहीं।

प्रश्न 5 (अथवा) • इकाई 3 10 अंक · ≈ 600–700 शब्द

ऑर्डर तालिका ORDER_LINE(OrderNo, OrderDate, CustID, CustName, CustCity, ItemNo, ItemName, UnitPrice, Qty) प्रत्येक ऑर्डर किए गए मद के लिए एक पंक्ति संग्रहीत करती है, जिसकी कुंजी (OrderNo, ItemNo) है। अ-सामान्यीकृत नमूना डेटा से आरंभ कर इसे चरणबद्ध रूप से 1NF, 2NF और 3NF तक सामान्यीकृत कीजिए, प्रयुक्त निर्भरताएँ और हटाई गई विसंगतियाँ बताइए, तथा सत्यापित कीजिए कि अपघटन हानिरहित है।

प्रस्तावना

सामान्यीकरण तालिका को उसकी प्रकार्यात्मक निर्भरताओं के अनुसार बाँटकर पुनरावृत्ति हटाता है। इस ऑर्डर तालिका की निर्भरताएँ हैं:

- OrderNo -> OrderDate, CustID
- CustID -> CustName, CustCity
- ItemNo -> ItemName, UnitPrice
- (OrderNo, ItemNo) -> Qty

चरण 0: अ-सामान्यीकृत डेटा

ऑर्डर 501 (5 अक्तूबर, ग्राहक C1 Meena, Raipur) में मद I1 Pen, मूल्य 10 (मात्रा 5) और I2 Notebook, मूल्य 40 (मात्रा 3) हैं। ऑर्डर 502 (6 अक्तूबर, C2 Karan, Durg) में I1 Pen (मात्रा 10) है। ऑर्डर 503 (8 अक्तूबर, C1 Meena, Raipur) में I3 Folder, मूल्य 25 (मात्रा 4) और I2 Notebook (मात्रा 2) हैं। प्रत्येक ऑर्डर एक रिकॉर्ड है जिसमें मदों का **दोहराव समूह (रिपीटिंग ग्रुप)** है, जो 1NF का उल्लंघन है।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

चरण 1: प्रथम सामान्य रूप

प्रत्येक कक्ष में एक परमाणुक मान होना चाहिए, इसलिए हर मद को अपनी पंक्ति मिलती है, और कुंजी (OrderNo, ItemNo) बनती है।

OrderNo	OrderDate	CustID	CustName	CustCity	ItemNo	ItemName	UnitPrice	Qty
501	05-Oct	C1	Meena	Raipur	I1	Pen	10	5
501	05-Oct	C1	Meena	Raipur	I2	Notebook	40	3
502	06-Oct	C2	Karan	Durg	I1	Pen	10	10
503	08-Oct	C1	Meena	Raipur	I3	Folder	25	4
503	08-Oct	C1	Meena	Raipur	I2	Notebook	40	2

विसंगतियाँ: Meena का नाम और शहर 5 में से 4 पंक्तियों में संग्रहीत है। यदि उसका शहर केवल कुछ पंक्तियों में बदला जाए तो अद्यतन विसंगति होती है। स्टैपलर जैसा नया मद तब तक जोड़ा नहीं जा सकता जब तक कोई उसे ऑर्डर न करे, क्योंकि कुंजी को OrderNo चाहिए। ऑर्डर 502 हटाने पर ग्राहक Karan का एकमात्र रिकॉर्ड भी हट जाता है, जो विलोपन विसंगति है।

चरण 2: द्वितीय सामान्य रूप

आंशिक निर्भरताएँ हैं क्योंकि अकेले OrderNo और ItemNo गैर-कुंजी गुणों को निर्धारित करते हैं। तालिका को इस प्रकार बाँटा जाता है कि प्रत्येक गैर-कुंजी गुण अपनी तालिका की पूरी कुंजी पर निर्भर हो।

- ORDERS(**OrderNo**, OrderDate, CustID, CustName, CustCity): 3 पंक्तियाँ
- ITEM(**ItemNo**, ItemName, UnitPrice): 3 पंक्तियाँ
- ORDER_LINE(**OrderNo, ItemNo**, Qty): 5 पंक्तियाँ

मद का विवरण अब एक ही बार संग्रहीत है, इसलिए Pen और उसका मूल्य केवल एक पंक्ति में है।

चरण 3: तृतीय सामान्य रूप

ORDERS में **संक्रमणीय निर्भरता** OrderNo -> CustID -> CustName, CustCity है। ग्राहक के तथ्य अलग तालिका में ले जाए जाते हैं।

- CUSTOMER में (C1, Meena, Raipur) और (C2, Karan, Durg) हैं।
- ORDERS में (501, 05-Oct, C1), (502, 06-Oct, C2) और (503, 08-Oct, C1) हैं।

अंतिम डिज़ाइन में चार तालिकाएँ हैं: **CUSTOMER(CustID, CustName, CustCity)**, **ORDERS(OrderNo, OrderDate, CustID)**, **ITEM(ItemNo, ItemName, UnitPrice)** और **ORDER_LINE(OrderNo, ItemNo, Qty)**। प्रत्येक तालिका में हर निर्धारक कुंजी है, इसलिए डिज़ाइन BCNF में भी है।

केवल 2NF पर क्यों न रुकें, 3NF तक क्यों जाएँ

चरण 2 के बाद भी तालिका ORDERS में Meena का नाम और शहर ऑर्डर 501 और ऑर्डर 503 दोनों के लिए संग्रहीत है। यदि वह Bilaspur चली जाए, तो दो पंक्तियाँ बदलनी पड़ेंगी, और छूटी हुई पंक्ति डेटाबेस को असंगत छोड़ देती है। केवल चरण 3 में प्रत्येक ग्राहक एक ही बार संग्रहीत होता है, इसलिए अद्यतन एक ही स्थान पर होता है। सामान्यीकरण तब पूर्ण होता है जब प्रत्येक गैर-कुंजी गुण कुंजी पर, पूरी कुंजी पर और केवल कुंजी पर निर्भर हो।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

हानिरहित जोड़ का सत्यापन

ORDER_LINE को ORDERS से OrderNo पर, ORDERS को CUSTOMER से CustID पर, और ORDER_LINE को ITEM से ItemNo पर जोड़कर नमूना डेटा पर चलाया गया। यह ठीक **5 मूल पंक्तियाँ** लौटाता है, न कोई अतिरिक्त न कोई कम, क्योंकि प्रत्येक जोड़ का स्तंभ जुड़ने वाली किसी तालिका की कुंजी है।

सामान्यीकरण के बाद विसंगतियाँ

- **अद्यतन:** Meena का शहर CUSTOMER की एक पंक्ति में बदला जाता है।
- **प्रविष्टि:** स्टैपलर बिना किसी ऑर्डर के ITEM में जोड़ा जाता है।
- **विलोपन:** ऑर्डर 502 हटाने पर ORDERS और ORDER_LINE की पंक्तियाँ हटती हैं, पर Karan CUSTOMER में बना रहता है।

मूल तालिका में 45 कक्ष थे और चार तालिकाओं में 39 हैं, और अधिक महत्वपूर्ण यह है कि प्रत्येक तथ्य एक ही बार संग्रहीत है।

निष्कर्ष

दोहराव समूह, फिर आंशिक निर्भरताएँ, फिर संक्रमणीय निर्भरताएँ हटाकर एकल तालिका को CUSTOMER, ORDERS, ITEM और ORDER_LINE में बाँटा गया। यह अपघटन हानिरहित है और अद्यतन, प्रविष्टि तथा विलोपन विसंगतियाँ हटाता है।

प्रश्न 6 • इकाई 4 10 अंक · ≈ 600–700 शब्द

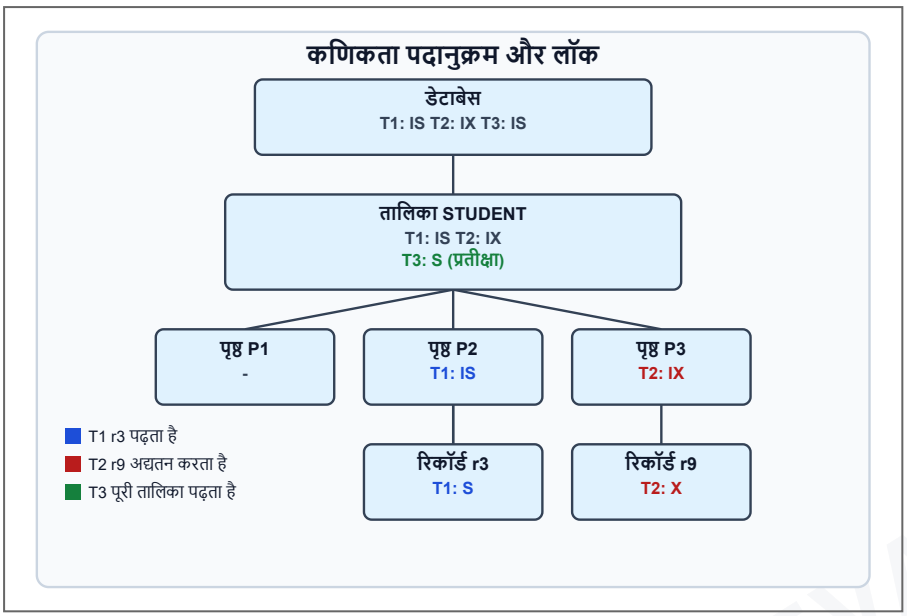
बहु-कणिकता लॉकिंग (मल्टीपल ग्रैन्युलैरिटी लॉकिंग) समझाइए। कणिकता पदानुक्रम, लॉक मोड IS, IX, S, SIX और X, संगतता आव्यूह और लॉकिंग नियमों का वर्णन कीजिए। उदाहरण से दिखाइए कि तीन ट्रांज़ैक्शन पदानुक्रम को कैसे लॉक करते हैं और किसे प्रतीक्षा करनी पड़ती है।

प्रस्तावना

सामान्य लॉकिंग में डेटा मद का एक ही निश्चित आकार रखा जाता है। यदि मद पूरी तालिका हो, तो एक पंक्ति अद्यतन करने वाला ट्रांज़ैक्शन बाकी सबको रोक देता है। यदि मद एक रिकॉर्ड हो, तो दस लाख रिकॉर्ड पढ़ने वाले ट्रांज़ैक्शन को दस लाख लॉक लेने पड़ते हैं। **बहु-कणिकता लॉकिंग** अनेक स्तरों पर लॉक की अनुमति देकर यह दुविधा हटाती है, ताकि बड़ा पाठक तालिका को एक बार लॉक करे और छोटा अद्यतनकर्ता केवल एक रिकॉर्ड लॉक करे।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

क. कणिकता पदानुक्रम



कणिकता पदानुक्रम और तीन ट्रांज़ैक्शनों द्वारा लिए गए लॉक

डेटा मद वृक्ष के रूप में सजाए जाते हैं: **डेटाबेस** में **तालिकाएँ** (फ़ाइलें) होती हैं, तालिका में **पृष्ठ (पेज)** और पृष्ठ में **रिकॉर्ड**। किसी नोड को लॉक करने पर उसके सभी वंशज अप्रत्यक्ष रूप से लॉक हो जाते हैं। इसलिए तालिका को S मोड में लॉक करने वाला ट्रांज़ैक्शन उसके नीचे के प्रत्येक पृष्ठ और रिकॉर्ड को बिना और लॉक के पढ़ सकता है।

ख. लॉक मोड

- **S (साझा):** नोड और उसके वंशज पढ़े जा सकते हैं।
- **X (अनन्य):** नोड और उसके वंशज पढ़े और लिखे जा सकते हैं।
- **IS (आशय साझा):** कोई वंशज S मोड में लॉक किया जाएगा।
- **IX (आशय अनन्य):** कोई वंशज X या S मोड में लॉक किया जाएगा।
- **SIX (साझा और आशय अनन्य):** पूरा उपवृक्ष S मोड में पढ़ा जाता है और कुछ वंशज X मोड में लॉक किए जाएँगे।

आशय लॉक पूर्वजों पर लगाए जाते हैं। वे बड़ा लॉक चाहने वाले ट्रांज़ैक्शन को चेतावनी देते हैं कि कोई नीचे काम कर रहा है, बिना हर वंशज की जाँच किए।

ग. संगतता आव्यूह

धारित / अनुरोधित	IS	IX	S	SIX	X
IS	हाँ	हाँ	हाँ	हाँ	नहीं
IX	हाँ	हाँ	नहीं	नहीं	नहीं
S	हाँ	नहीं	हाँ	नहीं	नहीं
SIX	हाँ	नहीं	नहीं	नहीं	नहीं
X	नहीं	नहीं	नहीं	नहीं	नहीं

घ. लॉकिंग नियम

1. लॉक को संगतता आव्यूह का पालन करना चाहिए।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

2. लॉकिंग वृक्ष की जड़ से शुरू होती है।
3. किसी नोड पर S या IS पाने के लिए ट्रांज़ैक्शन के पास उसके जनक पर IS या IX होना चाहिए।
4. X, IX या SIX पाने के लिए जनक पर IX या SIX होना चाहिए।
5. लॉक द्वि-चरण लॉकिंग का पालन करते हैं, और पत्ती से जड़ की ओर छोड़े जाते हैं।

ङ. उदाहरण

तालिका STUDENT लीजिए जिसके पृष्ठ P1, P2 और P3 हैं।

ट्रांज़ैक्शन	संक्रिया	क्रम से माँगे गए लॉक
T1	पृष्ठ P2 में रिकॉर्ड r3 पढ़ना	डेटाबेस पर IS, STUDENT पर IS, P2 पर IS, r3 पर S
T2	पृष्ठ P3 में रिकॉर्ड r9 अद्यतन करना	डेटाबेस पर IX, STUDENT पर IX, P3 पर IX, r9 पर X
T3	पूरी तालिका STUDENT पढ़ना	डेटाबेस पर IS, फिर STUDENT पर S

- **T1 और T2 साथ चलते हैं।** डेटाबेस और STUDENT पर IS और IX संगत हैं, और वे अलग पृष्ठों और रिकॉर्डों का प्रयोग करते हैं, इसलिए किसी को प्रतीक्षा नहीं करनी पड़ती।
- **T3 को प्रतीक्षा करनी पड़ती है।** यह डेटाबेस पर IS ले सकता है, पर STUDENT पर S, T2 के IX से टकराता है, क्योंकि आव्यूह में S और IX के लिए नहीं है। T3 तभी आगे बढ़ता है जब T2 अपने लॉक छोड़ दे, और T1 से इसका कोई टकराव नहीं होता, क्योंकि S और IS संगत हैं।

यदि T3 तालिका पढ़ना और कुछ पंक्तियाँ अद्यतन करना चाहता, तो वह STUDENT पर SIX और फिर उन रिकॉर्डों पर X माँगता। SIX में अन्य ट्रांज़ैक्शन केवल IS रख सकते हैं, इसलिए वे अलग-अलग रिकॉर्ड फिर भी पढ़ सकते हैं।

च. लाभ और सीमाएँ

- छोटे ट्रांज़ैक्शनों को उच्च समवर्तिता मिलती है, जबकि बड़े ट्रांज़ैक्शनों को कुछ ही लॉक चाहिए, जिससे लॉक तालिका का भार घटता है।
- DBMS स्तर स्वयं चुन सकता है, जैसे अनेक रिकॉर्ड लॉक होने के बाद रिकॉर्ड लॉक से तालिका लॉक पर चला जाना (लॉक एस्केलेशन)।
- प्रोटोकॉल अधिक जटिल है, और गतिरोध फिर भी हो सकते हैं, इसलिए गतिरोध की पहचान या रोकथाम फिर भी आवश्यक है।

निष्कर्ष

बहु-कणिकता लॉकिंग डेटा को वृक्ष में सजाती है और पूर्वजों पर आशय लॉक का प्रयोग करती है, ताकि भिन्न आकार के ट्रांज़ैक्शन डेटाबेस को सुरक्षित रूप से साझा कर सकें। उदाहरण में T1 और T2 समानांतर चलते हैं और तालिका पढ़ने वाला T3 T2 की प्रतीक्षा करता है।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

प्रश्न 6 (अथवा) • इकाई 4 10 अंक · ≈ 600–700 शब्द

लॉग-आधारित पुनःप्राप्ति समझाइए: राइट-अहेड लॉग, आस्थगित और तात्कालिक डेटाबेस संशोधन, तथा चेकपॉइंट। उत्तर में दिया गया लॉग लिखे जाने के बाद सिस्टम क्रैश हो जाता है। दोबारा किए जाने (रीडू) और रद्द किए जाने (अनडू) वाले ट्रांज़ैक्शन तथा डेटा मदों के अंतिम मान ज्ञात कीजिए।

प्रस्तावना

ट्रांज़ैक्शन के बीच में क्रैश हो सकता है, जिससे डेटाबेस में कुछ अद्यतन लिखे रह जाते हैं और कुछ खो जाते हैं। ACID के **परमाणुकता** और **स्थायित्व** भाग पुनःप्राप्ति से मिलते हैं, और सबसे प्रचलित विधि **लॉग** रखती है, जो स्थिर भंडारण पर क्रमिक फ़ाइल है और डेटाबेस में लागू करने से पहले हर अद्यतन को दर्ज करती है।

क. लॉग और राइट-अहेड नियम

अद्यतन के लॉग रिकॉर्ड का रूप (ट्रांज़ैक्शन, डेटा मद, पुराना मान, नया मान) होता है। अन्य रिकॉर्ड ट्रांज़ैक्शन का आरंभ, कमिट या एबॉर्ट दर्शाते हैं। **राइट-अहेड लॉगिंग (WAL) नियम** कहता है कि अद्यतन का लॉग रिकॉर्ड, बदला हुआ डेटा पृष्ठ डेटाबेस में लिखे जाने से पहले स्थिर भंडारण में पहुँचना चाहिए, और ट्रांज़ैक्शन तभी कमिट माना जाता है जब उसका कमिट रिकॉर्ड लॉग में लिखा जा चुका हो। पुराना मान अनडू और नया मान रीडू की सुविधा देता है।

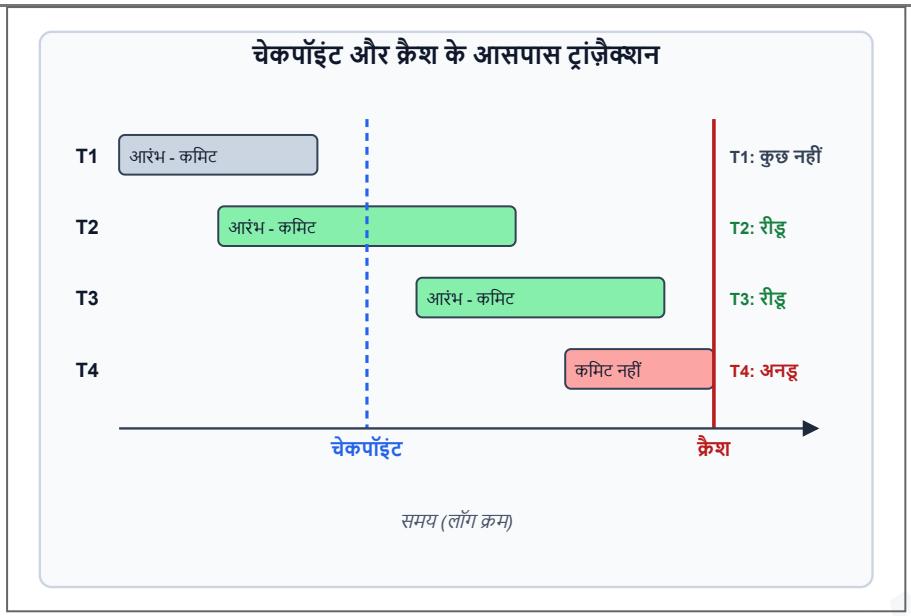
ख. आस्थगित और तात्कालिक संशोधन

बिंदु	आस्थगित संशोधन	तात्कालिक संशोधन
डेटाबेस कब बदलता है	कमिट रिकॉर्ड लॉग होने के बाद ही	ट्रांज़ैक्शन के दौरान कभी भी
क्रैश के बाद आवश्यक संक्रिया	केवल रीडू	अनडू और रीडू दोनों
लॉग में पुराना मान चाहिए	नहीं	हाँ
डिस्क पर अकमिट डेटा	कभी नहीं	संभव

ग. चेकपॉइंट

बिना सहायता के पुनःप्राप्ति को पूरा लॉग पढ़ना पड़ता। **चेकपॉइंट** पर सिस्टम सभी लॉग रिकॉर्ड और सभी संशोधित बफ़र डिस्क पर लिखता है, फिर एक चेकपॉइंट रिकॉर्ड लिखता है जो उस क्षण सक्रिय ट्रांज़ैक्शनों की सूची देता है। क्रैश के बाद केवल अंतिम चेकपॉइंट से आगे का लॉग और उस समय सक्रिय ट्रांज़ैक्शन ही ध्यान माँगते हैं।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।



चेकपॉइंट और क्रैश के आसपास चार ट्रांज़ैक्शनों की समय-रेखा

घ. हल किया गया उदाहरण

प्रारंभिक मान A = 1000, B = 2000, C = 700 और D = 500 हैं। क्रैश के समय लॉग है:

क्र.	लॉग रिकॉर्ड
1	T1 start
2	T1, A, 1000, 900
3	T2 start
4	T2, B, 2000, 2300
5	T1 commit
6	Checkpoint, सक्रिय ट्रांज़ैक्शन: T2
7	T3 start
8	T3, C, 700, 650
9	T2 commit
10	T4 start
11	T4, D, 500, 400
12	T3 commit
-	**क्रैश**

सूचियाँ बनाना। अनडू सूची में चेकपॉइंट पर सक्रिय ट्रांज़ैक्शन, यहाँ T2, और रीडू सूची रिक्त से आरंभ कीजिए। चेकपॉइंट से आगे स्कैन कीजिए। T3 आरंभ होता है और अनडू सूची में जुड़ता है; T2 कमिट होता है और रीडू सूची में जाता है; T4 आरंभ होता है और अनडू सूची में जुड़ता है; T3 कमिट होता है और रीडू सूची में जाता है। अंतिम सूचियाँ **रीडू = T2, T3** और **अनडू = T4** हैं। T1 चेकपॉइंट से पहले कमिट हो चुका था, इसलिए उसका अद्यतन पहले ही डिस्क पर पहुँच चुका है और उसे किसी कार्रवाई की आवश्यकता नहीं।

पुनःप्राप्ति की कार्रवाइयाँ।

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

1. **T4 को अनडू करना:** लॉग को पीछे की ओर स्कैन कर D को 400 से उसके पुराने मान 500 पर लौटाइए, फिर T4 के लिए एबॉर्ट रिकॉर्ड लिखिए।
2. **T2 और T3 को रीडू करना:** चेकपॉइंट से आगे स्कैन कर नए मान दोबारा लागू कीजिए, ताकि C, यदि खो गया हो, फिर 650 हो जाए। T2 का B अद्यतन चेकपॉइंट से पहले था, इसलिए वह 2300 के रूप में पहले से डिस्क पर है।

अंतिम मान: A = 900, B = 2300, C = 650 और D = 500। कमिट हुए ट्रांज़ैक्शन T1, T2 और T3 पूरी तरह प्रतिबिंबित हैं, और अकमिट ट्रांज़ैक्शन T4 ने कोई निशान नहीं छोड़ा, यही परमाणुकता और स्थायित्व है।

कमिट रिकॉर्ड का प्रभाव। यदि क्रैश रिकॉर्ड 12 से ठीक पहले होता, तो T3 का कोई कमिट रिकॉर्ड न होता, इसलिए वह T4 के साथ अनडू सूची में आता और C अपने पुराने मान 700 पर लौटाया जाता। केवल कमिट रिकॉर्ड तय करता है कि ट्रांज़ैक्शन बचेगा या नहीं, इसीलिए उपयोगकर्ता को ट्रांज़ैक्शन पूर्ण बताने से पहले उसे डिस्क पर बलपूर्वक लिखा जाता है।

निष्कर्ष

लॉग-आधारित पुनःप्राप्ति पहले लॉग लिखने पर निर्भर है। आस्थगित संशोधन में केवल रीडू चाहिए, तात्कालिक संशोधन में अनडू और रीडू, और चेकपॉइंट यह सीमित करते हैं कि लॉग का कितना भाग स्कैन हो। उदाहरण में T2 और T3 रीडू होते हैं, T4 अनडू होता है और डेटाबेस संगत अवस्था में लौट आता है।

ONLINE STUDENT SEVA · PARIKSHA PASS

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

How to use this Demo Set इस डेमो सेट का उपयोग कैसे करें

1. This Demo Set follows the same pattern, marks and layout as the real paper.
2. Cover the answer, write your own attempt first, then compare it with the model answer.
3. Short answers (100-120 words) need a definition, two or three points and a small example, table or SQL line.
4. For 10-mark answers use the structure: introduction, headed points, a diagram or table, example, conclusion.
5. In design and normalization questions write the given dependencies first, then the steps, and show the final tables clearly.
6. Draw neat labelled E-R diagrams and decomposition trees; they earn marks and save words.
7. These are practice questions only and are not taken from the actual sets.

1. यह डेमो सेट असली पेपर के समान पैटर्न, अंक और प्रारूप का है।
2. उत्तर ढककर पहले स्वयं लिखें, फिर आदर्श उत्तर से मिलान करें।
3. लघु उत्तर (100-120 शब्द) में परिभाषा, दो-तीन बिंदु और छोटा उदाहरण, तालिका या SQL पंक्ति होनी चाहिए।
4. 10 अंक के उत्तर में प्रस्तावना, शीर्षकों सहित बिंदु, आरेख या तालिका, उदाहरण और निष्कर्ष की संरचना रखें।
5. डिज़ाइन और नॉर्मलाइज़ेशन के प्रश्नों में पहले दी गई निर्भरताएँ लिखें, फिर चरण, और अंत में अंतिम तालिकाएँ साफ़ दिखाएँ।
6. साफ़, नामांकित E-R आरेख और अपघटन वृक्ष बनाएँ; इनसे अंक मिलते हैं और शब्द बचते हैं।
7. ये केवल अभ्यास प्रश्न हैं और वास्तविक सेटों से नहीं लिए गए हैं।

ONLINE STUDENT SEVA · PARIKSHA PASS

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।

Unlock the full sets in the Pariksha Pass App पूरे सेट परीक्षा पास ऐप में अनलॉक करें

You have just worked through one complete Demo Set of **Relational Database Management System** (CASC-08T) in the pattern of your university paper: 10 objective questions, 5 short-answer questions and 4 long-answer questions with internal choice, each with a model answer in English and Hindi.

The questions in this Demo Set are **practice questions chosen for demonstration**. They are not taken from the actual Guessing Paper sets.

The full Guessing Paper sets for your university, course and subject – Set A (Most Probable), Set B (Highly Likely) and Set C (Alternate) – are available in the Pariksha Pass App. They follow the same pattern, structure and answer style as this Demo Set.

आपने अभी अपने विश्वविद्यालय के प्रश्न-पत्र के प्रारूप में **रिलेशनल डेटाबेस मैनेजमेंट सिस्टम** का एक पूरा डेमो सेट हल किया है: 10 वस्तुनिष्ठ प्रश्न, 5 लघु उत्तरीय प्रश्न और आंतरिक विकल्प सहित 4 दीर्घ उत्तरीय प्रश्न, सभी के आदर्श उत्तर अंग्रेज़ी और हिंदी में।

इस डेमो सेट के प्रश्न प्रदर्शन के लिए चुने गए अभ्यास-प्रश्न हैं। ये असली गेसिंग पेपर सेटों में से नहीं लिए गए हैं।

आपके विश्वविद्यालय, कोर्स और विषय के पूरे गेसिंग पेपर सेट - सेट A (सर्वाधिक संभावित), सेट B (अत्यधिक संभावित) और सेट C (वैकल्पिक) - परीक्षा पास ऐप में उपलब्ध हैं। इनका प्रारूप, संरचना और उत्तर-शैली इसी डेमो सेट जैसी है।

Online Student Seva · Pariksha Pass App · DEMO SAMPLE

ONLINE STUDENT SEVA · PARIKSHA PASS

Download Pariksha Pass App >

Get all 3 full Guessing Paper sets on Google Play

© Online Student Seva · Exclusive property · Accessible only through the Pariksha Pass App · Copying, sharing or selling is strictly prohibited and will invite serious legal action.
© ऑनलाइन स्टूडेंट सेवा · विशेष संपत्ति · केवल परीक्षा पास ऐप पर उपलब्ध · कॉपी, शेयर या बिक्री पर कड़ी कानूनी कार्रवाई होगी।