

Database Management System

[Syllabus: BPSC CS: Definition of DBMS, types of DBMS, its advantages and disadvantages, Data model: ER model and relational model. Integrity constraints. Functional dependencies. Assertions and triggers. File organization: Definition of various file organization, classification and Representation. Indexing techniques: sparse and dense indexing. B+ tree indexing, hash indexing. Relational database design: normalization, 2NF, 3NF and BCNF. Query processing: Various notations, cost estimation of selection operation and join operation. Transaction concept and concurrency control: Lock based protocol, deadlock handling. SQL and application using SQL.

NTRCA CS: Database system concept, Data abstraction; Database languages: Data management; Database system structure. Database model, Relational algebra and SQL. Database design, Indexing, Normalization.

NTRCA ICT (BM): ডাটাবেজ ম্যানেজমেন্ট, ডাটাবেজ ডিজাইন ও ডিজাইন টেবিল।

NTRCA ICT: Introduction to Database Systems: Evolution of file processing systems, role of databases in organizations, core components of a database environment; Data Modeling: the Entity-Relationship Diagram and its symbols and constructs; The Relational Model and Normalization: relational model, normalization, transformation of an entityrelationship data diagram into a relational model; SQL - A Standard Navigation Language for Relational.

Databases; Overview of Object-Oriented Databases: object-oriented data model, implementation of object persistence using relational databases. Indexing and Hashing: Basic Concepts, Ordered Indices, B+-Tree Index Files, B-Tree Index Files, Static Hashing, Dynamic Hashing, Comparison of Ordered Indexing and Hashing; Query Processing: Overview, Measures of Query Cost, Selection Operation, Sorting, Join Operation, Evaluation of Expressions; Query Optimization: Introduction, Transformation of Relational Expressions, Catalog Information for Cost Estimation, Statistical Information for Cost Estimation, Cost-based optimization; Transactions: Transaction Concept, Transaction State, Concurrent Executions, Serializability; Concurrency Control: Lock-Based Protocols, Timestamp-Based Protocols; Recovery System: Failure Classification, Storage Structure, Recovery and Atomicity, Log-Based Recovery, Recovery With Concurrent Transactions; Data Analysis and Mining: Data Mining, Decision tree, Bayes theory, Randomize tree; Database System Architectures: Centralized and Client-Server Systems, Server System Architectures, Parallel Systems, Distributed Systems, Network Types; Parallel Databases: Introduction, I/O Parallelism, Interquery Parallelism, Intraquery Parallelism, Intraoperation Parallelism, Interoperation Parallelism; Distributed Databases: Heterogeneous and Homogeneous Databases: Distributed Data Storage, Distributed Transactions, Commit Protocols; Additional should be included: Database Design, Database Tuning Security and Authorization, Multidimensional query.]

Q Database Management System

প্রশ্ন ১. Data কলতে কি বুঝ? ইনফরমেশন কি? What do you mean by data? What is information?***

উত্তর: Data হল এ সমষ্টি উপাদান বা উপকরণ যাদের প্রক্রিয়া করলে Information (তথ্য) পাওয়া যায়। ইনফরমেশন ভাটা কে প্রক্রিয়া করলে যে অন্তর্ফুট পাওয়া যায়, তাকে ইনফরমেশন বলে।

উদাহরণ: data:- ১০, কলম, কেজিবিল etc. ইনফরমেশন- "দে পরীক্ষাতে ১০(data) মার্ক পেছেহে"

প্রশ্ন ২. মেটা ভাটা (Meta Data) কলতে কি বুঝ?
উত্তর: ভাটা সম্পর্কিত ভাটাকে মেটা ভাটা বলে। অর্থাৎ ভাটা সমূহ একে অপরের সাথে কিভাবে সম্পর্কৃত থাকে, তা যে ভাটা এর মাধ্যমে কৈবল্য করা হয় তাকে মেটা ভাটা বলে।

উদাহরণ: Image → image height,image width, resolution etc.

প্রশ্ন ৩. ভাটা কত প্রকার ও কি কি? How many types of data and what?***

উত্তর: ভাটা প্রধানত ৩ প্রকার। যথাঃ

- a) Numerical Data
- b) Textual Data
- c) Graphical Data

এছাড়াও আরো কয়েকটি টাইপের ভাটা Database এ ব্যবহার হয়।

- a) Logical Data
- b) Currency
- c) Data & Time
- d) Memotype

প্রশ্ন ৪. ভাটাবেস কলতে কি বুঝ? What do you mean by database? [BKKB - 2017, Dept of ICT - 2014, Ministry of Finance-2013, BRTA-2012] ***

উত্তর: কম্পিউটার সিস্টেম এ জামাকৃত (Store) সর্বমোট structure Data এর সমষ্টিকে ভাটাবেস বলে।

প্রশ্ন ৫. What are the uses of the database in-office and use Database in office administration? অফিসে ভাটাবেসের ব্যবহার এবং অফিস প্রশাসনে ভাটাবেসের ব্যবহার কী?

Answer: প্রধানত, ভাটাবেস সংস্কার্যার ভাটাবেসের তথ্য সুরক্ষিত করতে এবং নির্ভুল এবং সামঞ্জস্যপূর্ণ নিশ্চিত করার কাজ করে। ভাটাবেস সংস্কার্যার এর অন্তর্ভুক্ত ফাঁকশন সমূহ:

- a) Data storage
- b) Data backup and recovery
- c) Data presentation and reporting
- d) Multi-user access control
- e) Data security management
- f) Database communication

প্রশ্ন ৬. What are the roles of Database Engineer? [ভাটাবেস ইঞ্জিনিয়ারের ভূমিকা কি?**]

- i. বিদ্যমান ভাটাবেস প্রোগ্রামগুলির কর্মক্ষমতা বজায় রাখা এবং উন্নত করা।
- ii. ভাটাবেস ডেভেলপমেন্ট টিমকে নতুন ভাটাবেস প্রোগ্রাম ডিজাইনে সহায়তা করা যা প্রতিটানের টেটা স্টোরেজের চাহিদা পূরণ করে।
- iii. ভাটাবেস এবং ভাটাবেস সম্পর্কিত সিস্টেমগুলির কর্মক্ষমতা নিশ্চিত করতে পর্যবেক্ষণ করা।
- iv. টেটা স্টোরেজ ক্ষমতা বাড়িয়ের জন্য নতুন support program এবং ক্লিন্ট লেখা।
- v. ভাটাবেস এবং ব্যবহারকারীর রিপোর্ট, সেইসাথে সিস্টেম তথ্য প্রযোচনা করা।
- vi. ভাটাবেস ক্লিন্ট এবং প্রোগ্রামগুলিতে ডিবাগিং পর্কটি সম্পাদন করা, সেইসাথে বড় সমাধান করা।
- vii. ভাটাবেস প্রশাসকদের পরামর্শ দেওয়া এবং তাদের প্রতিক্রিয়া সহায়তা প্রদান করা।
- viii. নিরাপদে সঞ্চয় ব্যাক আপ এবং টেটা সংরক্ষণের সর্বোত্তম অনুরীলনগুলি মেনে চলা।
- ix. ভাটাবেস ডিজাইন, কনফিগারেশন এবং কর্মক্ষমতা সম্পর্কিত নথিভুক্তকরণ প্রক্রিয়া চালু করা।

প্রশ্ন ৭. টেটাবেজের উপাদান কলো আলোচনা করুন। [Discuss the components of the database?]

অ্যাট্রিবিট/ফিল্ড/কলম : কোনো একটি এন্টিটি সেটের যে প্রোপের্টিজগুলো ঐ এন্টিটির বৈশিষ্ট্যগুলো প্রকাশ করে এবং যার পের ডিটি করে উপাত্ত গ্রহণ, প্রক্রিয়াকরণ ও সংরক্ষণ করা হয় তাকে অ্যাট্রিবিট বলে। যেমন- Students একটি এন্টিটি যার অ্যাট্রিবিট হলো Id, Name ইত্যাদি। অ্যাট্রিবিটকে ডিজুন্যাল টেটাবেজের প্রত্যামে সাধারণত টেটা কিন্তু বলে।

অ্যাট্রিবিট ভেলু: একটি এন্টিটি সেটের প্রত্যেকটি অ্যাট্রিবিটের একটি নির্দিষ্ট মান আছে। অ্যাট্রিবিটের এ মানকে তার ভেলু বলে।

এন্টিটিসেট/রেকর্ড/চাপল/সারি: পরম্পর সম্পর্কযুক্ত একাধিক ফিল্ড নিয়ে গঠিত হয় এক একটি রেকর্ড।

এন্টিটিসেট/টেবিল এক বা একাধিক রেকর্ড নিয়ে টেবিল তৈরি হয়।

প্রশ্ন ৮.. DBMS কলতে কি বুঝ? What is DBMS? [BKKB-2017, 40 BCS, BPSC(ANE)-2020]

উত্তর: কম্পিউটার সিস্টেম এ জামাকৃত সর্বমোট স্ট্রাকচারত ভাটা সমষ্টিকে ভাটাবেস বলে। আর Store কৃত Data কে একসেস করার জন্য ব্যবহৃত এক সেট প্রোগ্রাম এর সমষ্টিকে ভাটাবেস ম্যানেজমেন্ট সিস্টেম বলে।

প্রশ্ন ৯. DBMS কত প্রকার ও কি কি? What is Different types of DBMS ? [৪০ BCS]***

উত্তর: DBMS কে প্রধানত ৪ ভাগে ভাগ করা হয়। যথাঃ

- a) Hierarchical DBMS
- b) Relational DBMS
- c) Network DBMS

প্রশ্ন ১০. What are the roles of Database Engineer? [**]

উত্তর: DBMS এর সুবিধা ও অসুবিধা সমূহ লিখুন Write advantage and disadvantage of DBMS?]

উত্তর: DBMS এর সুবিধা সমূহ:

- ক) DBMS এ ভাটা প্রবেশ অভ্যন্ত সহজ।
- খ) এতে টেটা স্টোরেজ স্পেস অনেক কম লাগে।
- গ) এতে ভাটা সিকিউরিটি বুবই বেশি।
- ঘ) এতে আ্যটোমেলিটি অসুবিধা নেই।
- ঙ) এতে শক্তিশালী ব্যাকআপ ও রিকভারি এর ব্যবহা আছে।
- ঁ) ইহা একটি স্ট্রাকচারড সিস্টেম।
- ঁঁ) ভাটা আইসোলেশন জনিত সমস্যা নেই।
- ঁঁঁ) এতে ভাটা কেন্দ্রীয়া/খোজ সহজ।
- ঁঁঁঁ) এতে ভাটা Manipulation সহজ।
- ঁঁঁঁঁ) এতে ভাটা ইন্টেগ্রেশন এর নিচয়তা থাকে।

DBMS এর অসুবিধা সমূহ:

- ক) এতে হার্ডওয়্যার ও সফটওয়্যার জনিত খরচ বেশী।
- খ) DBMS এ ভাটা ব্যাকআপ ও রিকভারি এর ব্যবহা জটিল।
- ঘ) এই সিস্টেমে ভাটা কেন্দ্রীয়তাবে নিয়ন্ত্রিত হয় বলে, কখনো কখনো বেশী অসুবিধার সৃষ্টি হয়।

প্রশ্ন ১১. DBMS এর কয়েকটি ব্যবহার/এপ্লিকেশন লিখ। [Write some application of DBMS?]***

উত্তর: DBMS এর কয়েকটি ব্যবহার/এপ্লিকেশন হলোঃ

- a) পে রোল (Pay Roll)
- b) মজুদকরণ (Inventory System)
- c) ক্রয়/বিক্রয় (Purchase/Sell)
- d) Real state Management
- e) Project Management
- f) Medical system
- g) Production system
- h) Cost Accounting
- i) Airline system
- j) Banking system
- k) Rail way system
- l) School/College/University Management
- m) Debit/Credit cost transaction

প্রশ্ন ১২. DBMS উদ্দেশ্য/লক্ষ্য/Purpose/প্রয়োজনীয়তা সমূহ লিখ। [Ministry of H & F-2014, Ministry of Finance-2013]***

উত্তর: DBMS এমন একটি সফটওয়্যার সিস্টেম যার দ্বারা ভাটাবেস এ জামাকৃত ভাটা সমূহকে সুস্কলেপ মানিপুলেট করা যায়। DBMS এর মাধ্যমিক উদ্দেশ্য রয়েছে। তা নিম্নরূপঃ-

- ঁ) ভাটা তে সহজ প্রবেশের ব্যবহা করাটি-(easy to access)
- ঁঁ) বিভিন্ন কারনে কনভেনশনাল ফাইল প্রসেসিং সিস্টেম এর অসুবিধা সূর করে ভাটাবেস ব্যবহারকারীগন যাতে অভ্যন্ত দ্রুত ও সহজে ভাটাবেস ম্যানেজমেন্ট সিস্টেম বলে।

প্রশ্ন ১৩. Conventional file processing system (CFPS)

এর বিভিন্ন সুবিধা ও অসুবিধা সমূহ লিখ।

উত্তর: CFPS এর বিভিন্ন সুবিধা ও অসুবিধা সমূহ নিম্নরূপঃ

CFPS এর সুবিধা সমূহঃ

- a) CFPS এ প্রয়োজনীয় হার্ডওয়্যার এর জটিলতা ও সফটওয়্যার এর ব্যবহা করত কম পড়ে।

- b) Backup & Recovery এর জটিলতা নেই।
 c) Data Manipulate করার জন্য বিশ্বাসীয় ভাবে ব্যবহার করে এপ্পুকেশন রচনা করা যায়।
 d) কেন্দ্রীয়ভাবে নির্যাতিত হয় না বলে, অনেক ফেরে কিছু সুবিধা পাওয়া যায়।

CFPS এর অসুবিধা সমূহ:
 a) এতে ব্যাকআপ ও রিকভারি এর ব্যবহা নেই।
 b) এতে ডাটা আইসোলেশন সমস্যা বিদ্যমান।
 c) এতে ডাটা আটোমেটিস্ট সমস্যা বিদ্যমান।
 d) এতে Inconsistency অবহা বিবাজ করে।
 e) এতে স্টোরেজ স্পেস বেশি লাগে।
 f) এটি একটি অনন্ত্রিকচারভ সিস্টেম।
 g) এতে ডাটা ইন্টিমিটি রক্ষা করা সম্ভব নয়।
 h) ইহার ডাটা একসেস প্রতিক্রিয়া করিন।

প্রশ্ন 18. Database Management system (DBMS) এবং Conventional File Processing System (CFPS) এর মধ্যে পার্থক্য কলো লিখ।*
 গতানুগতিক ফাইল সিস্টেম অপেক্ষা রিলেশনাল ডাটাবেস সিস্টেমের সুবিধাগুলি কি কি?

উত্তর: CFPS ও DBMS পার্থক্য সমূহ নিম্নরূপ-

CFPS	DBMS
১। যে ফাইল প্রসেসিং সিস্টেম এ কোন DBMS ব্যবহার না করে অন্যান্য গতানুগতিক ফাইল সিস্টেম ব্যবহার করা হয় তাকে CFPS বলে।	১। যে সফটওয়্যার সিস্টেম এর সাহায্যে ডাটাবেস এ জমান্ত ডাটা কে সুবিধাজনক ভাবে ও দক্ষতার সাথে Manipulate করা যায় তাকে DBMS বলে।
২। এতে ফাইল সমূহ বিভিন্ন ফর্ম্যাট এ থাকে।	২। এতে ফাইল সমূহ একই ফর্ম্যাট এ থাকে।
৩। এতে স্টোরেজ স্পেস বেশী লাগে।	৩। এতে স্টোরেজ স্পেস কম লাগে।
৪। ডাটা সিকিউরিটি নেই বললেই চলে।	৪। এতে ডাটা সিকিউরিটি ব্যবহা খুব ভালো।
৫। ডাটা আইসোলেশন জনিত সমস্যা বিদ্যমান।	৫। ডাটা আইসোলেশন জনিত সমস্যা নেই।
৬। এতে হার্ডওয়ার ও সফটওয়ার জনিত খরচ কম।	৬। এতে হার্ডওয়ার ও সফটওয়ার জনিত খরচ বেশি।
৭। এতে ডাটা ব্যাকআপ ও রিকভারি এর ব্যবহা নেই।	৭। এতে ডাটা ব্যাকআপ ও রিকভারি এর ব্যবহা আছে।
৮। এটি একটি অনন্ত্রিকচারভ সিস্টেম।	৮। এটি একটি স্ট্রোকচারভ সিস্টেম।
৯। এতে কেন্দ্রীয় ভাবে নির্যাতন ব্যবহা নেই।	৯। এতে কেন্দ্রীয়ভাবে নির্যাতন ব্যবহা আছে।
১০। ডাটা সমূহ শেয়ার করা যায় না।	১০। ডাটা সমূহ শেয়ার করা যায়।

প্রশ্ন 15. কয়েকটি DBMS প্যাকেজের নাম লিখ।**
উত্তর: DBMS এর কয়টি প্যাকেজের নাম হলো:

- a) MS Access
- b) MySQL
- c) SQL Server
- d) Oracle
- e) Jasmine
- f) Dbase
- g) FoxPro
- h) Sybase
- i) Foxbase
- j) Informix
- k) Ingrase
- l) IBMs DM2
- m) Microsoft SQL server

প্রশ্ন 16. Data Dictionary: DDL গুলোকে compilation করলে একটি set of table পাওয়া যায়। এদেরকে একটি বিশেষ ফাইলে জমা করা হয়। এই ফাইল টিকে ডাটা ডিকশনারী বলে।

প্রশ্ন 17. সাধারণ অপারেটিং সিস্টেম এ ফাইল হিসেবে কোন ডাটা জমা না রেখে ডাটাবেজ সিস্টেমে কেন রাখা হয়? Why is it kept in the database system without storing any data as a file in normal operating system?

উত্তর: সাধারণ অপারেটিং সিস্টেম এর ফাইল হিসেবে কোন ডাটা জমা না রেখে ডাটাবেজ সিস্টেমে রাখা হয় কারণ:

- DBMS provides data independence
- DBMS provides efficient access of data,
- Faster retrieval of data
- DBMS provides indexing and query optimization mechanisms
- Data integrity and security:
- DBMS provides concurrency control.
- Crash Recovery: DBMS maintains log files and has mechanisms which recover the system to a consistent state after a crash.

প্রশ্ন 18. Classify the failure of Database Management ? | ডেটাবেজ ম্যানেজমেন্টের ব্যর্থাকে শ্রেণীবর্ত করা?|

Answer: Failure Classification of database management system is three types.

- a) Transaction failure
- b) System crash
- c) Disk failure

(a) লেনদেন ব্যর্থা(Transaction failure): লেনদেন ব্যর্থা তখন ঘটে যখন এটি লেনদেন কার্যকর করতে বার্থ হয় বা যখন এটি এমন একটি স্থানে পৌছায় যেখান থেকে আর লেনদেন করা যায় না। যদি কিছু লেনদেন বা প্রক্রিয়া ব্যাহত হয় তাহলে একে লেনদেন ব্যর্থা বলা হয়। লেনদেন ব্যর্থার দুটি কারণ হতে পারে:

(i) মৌলিক ত্রুটি(Logical errors): যদি কিছু কোড ত্রুটি (code error) বা অভ্যন্তরীণ ত্রুটির (internal error condition) কারণে একটি লেনদেন সম্পূর্ণ না হয়, তাহলে মৌলিক ত্রুটি ঘটে।

(ii) সিনটাক্স ত্রুটি(Syntax error): এটি ঘটে যেখানে DBMS নিজেই একটি সক্রিয় লেনদেন বক করে দেখ কারণ ডাটাবেস সিস্টেম এটি কার্যকর করতে সক্ষম হ্যান্ড না।

(b) সিস্টেম ত্রুটি: প্রায়ত্ব ব্যর্থা বা অন্যান্য হার্ডওয়্যার বা সফটওয়্যার ব্যর্থার কারণে সিস্টেম ব্যর্থা ঘটতে পারে। উদাহরণ: অপারেটিং সিস্টেম ত্রুটি।

(c) ডিস্ক ব্যর্থা (Disk Failure): এটি ঘটে যখন হার্ড-ডিস্ক ড্রাইভ ক্ষেত্রে ড্রাইভগুলি প্রায়শই ব্যর্থ হয়। ডিস্ক ব্যর্থা খারাপ সেটের গঠন, কিংবা ডেক্স এবং ডেকে নাগালযোগাতা বা অন্য কোন ব্যর্থার কারণে ঘটে যা সমস্ত বা ডিস্ক স্টোরেজের কিছু অংশ হ্যান্ড করে।

৪. বিভিন্ন প্রকার ম্যানেজমেন্ট সিস্টেম।

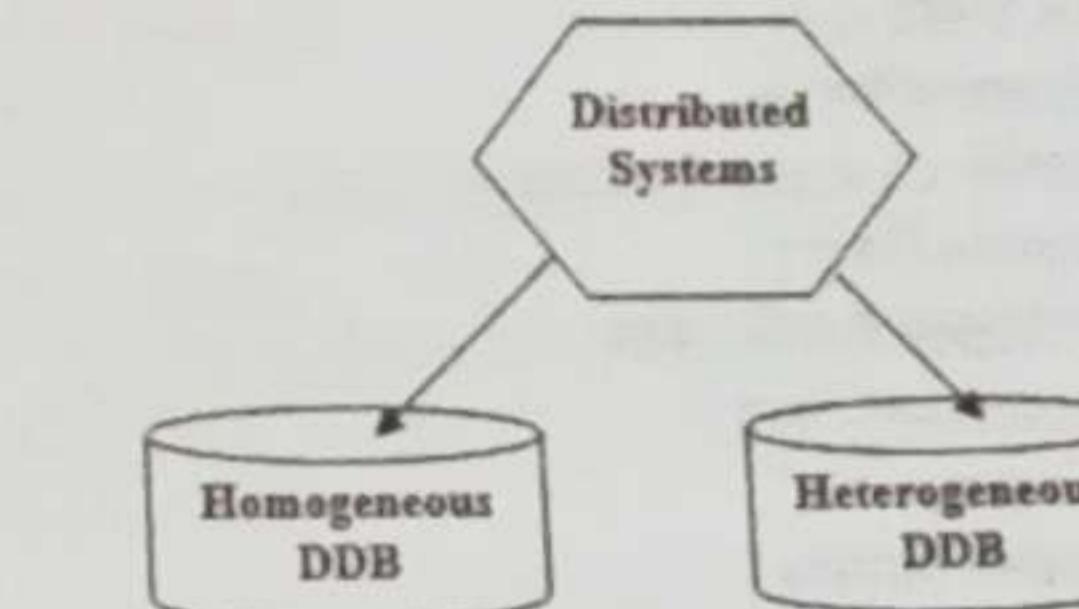
প্রশ্ন 1. সেন্ট্রালাইজড ডাটাবেস সিস্টেম কি? [What is a centralized database system?]

উত্তর: যে ডাটাবেস সিস্টেমে একটি মাত্র কম্পিউটার চলতে সক্ষম এবং এই সময়ে শুধুমাত্র একজন ব্যবহারকারীই প্রয়োজনীয় সিস্টেম কে ব্যবহার করতে পারে। অন্য কোন কম্পিউটার সিস্টেম এর সাথে ইন্টারয়েক্ট করতে পারে না, তাকে সেন্ট্রালাইজড ডাটাবেস সিস্টেম বলে।

গিপোর্ট ও ট্র্যাক করা যায় এবং রেকর্ডসমূহকে পরিবর্তন করা যায় তাকে ওয়েবে এনাবল ডেটাবেজ বলে।

প্রশ্ন 8. ডিস্ট্রিবিউটেড ডাটাবেজ সিস্টেম কি? [What is a distributed database system?]

উত্তর: একটি ডাটাবেস সিস্টেমের ডাটা সমূহ যখন দুই বা ততোধিক individual সিস্টেমে data store, insert, update ইত্যাদি করে তখন উক্ত ডাটাবেস সিস্টেমকে ডিস্ট্রিবিউটেড ডাটাবেজ সিস্টেম বলা হয়। ডিস্ট্রিবিউটেড ডাটাবেজ সিস্টেম এর individual সিস্টেম সমূহ তাদের inter communication line অথবা internet এর মাধ্যমে যোগাযোগ প্রতিষ্ঠা করে। ডিস্ট্রিবিউটেড ডাটাবেজ সিস্টেমে ব্যবহারকারীদের সহজেই ডেটা অ্যাক্সেস করতে পারে। ডিস্ট্রিবিউটেড ডাটাবেজ সিস্টেম ২ ভাগে বিভক্ত : যথা



■ Homogeneous DDB: Homogeneous ডিস্ট্রিবিউটেড ডাটাবেজ সিস্টেম একই অপারেটিং সিস্টেমে এ Run করে একই অ্যাপ্রুকেশন process এবং একই হার্ডওয়্যার ব্যবহার করে।

■ Heterogeneous DDB: Heterogeneous ডিস্ট্রিবিউটেড ডাটাবেজ সিস্টেম ভিন্ন অপারেটিং সিস্টেমে এ Run করে ভিন্ন অ্যাপ্রুকেশন process এবং ভিন্ন ভিন্ন হার্ডওয়্যার ব্যবহার করে।

ডিস্ট্রিবিউটেড ডাটাবেজ সিস্টেম এর ফিচার সমূহ : ডিস্ট্রিবিউটেড ডাটাবেজ সিস্টেম এর ফিচার সমূহ নিম্নরূপ-

- ক) লোকেশন ইভিপেন্ডেট।
- খ) ডিস্ট্রিবিউটেড কুয়েরী প্রসেস।
- গ) ডিস্ট্রিবিউটেড transaction management.
- ঘ) অপারেটিং সিস্টেম ইভিপেন্ডেট।
- ঙ) নেটওয়ার্ক ইভিপেন্ডেট।
- ছ) Transaction transparency.
- জ) ডিবিএমএস ইভিপেন্ডেট।
- ঝ) হার্ডওয়্যার ইভিপেন্ডেট।

প্রশ্ন 5. ডিস্ট্রিবিউটেড ডাটাবেস সিস্টেমের সুবিধা ও অসুবিধা লিখুন? [List the advantages and disadvantages of distributed database systems.]

ডিস্ট্রিবিউটেড ডাটাবেস সিস্টেম এর সুবিধা ও অসুবিধা হলোঃ

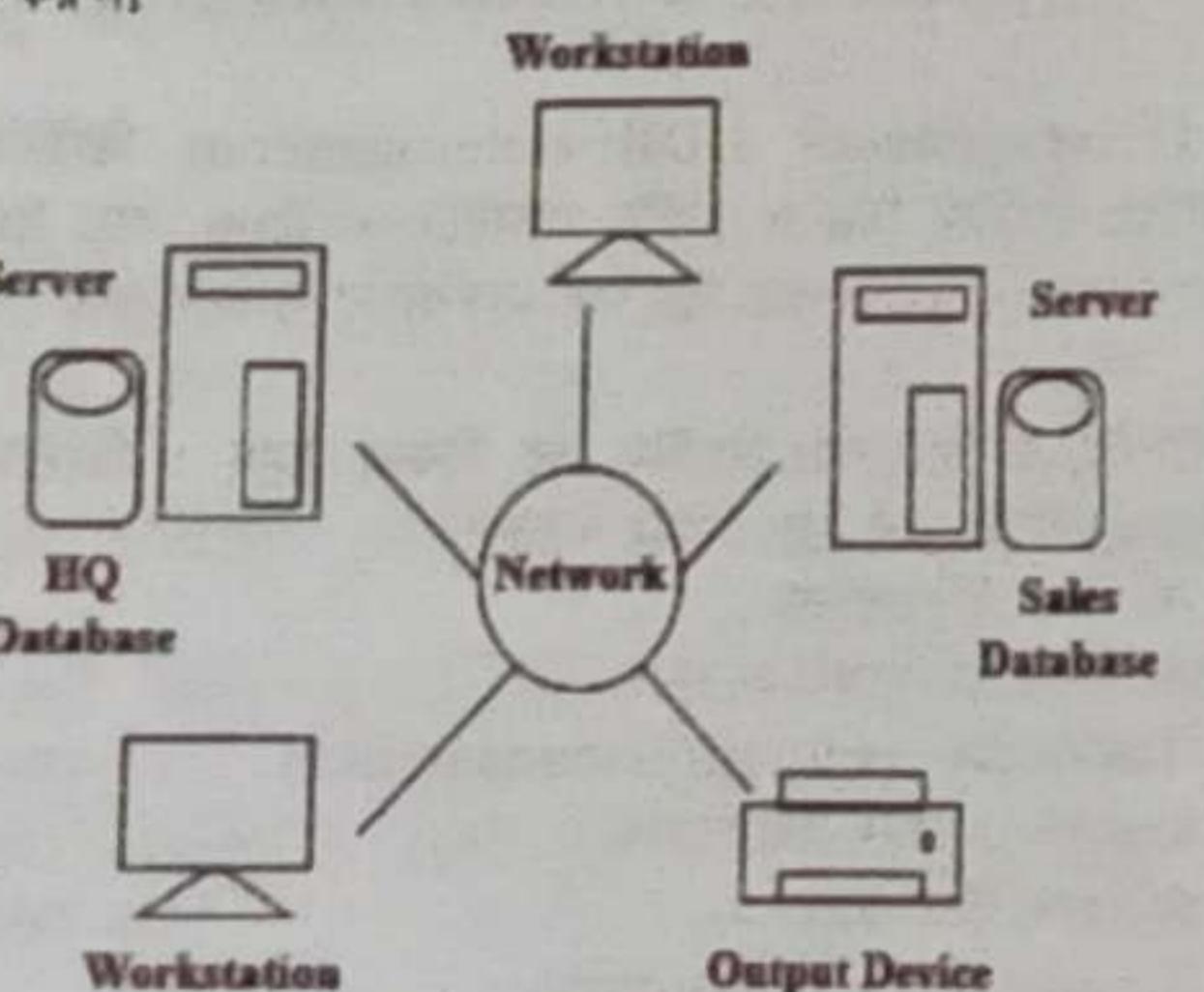
সুবিধাঃ
 ক) ইউজার যে কোন ডাটাবেজ এ প্রবেশ করে তা একেস করতে পারে।
 খ) এতে অটোমিষ সুবিধা প্রদান করে।
 গ) ডাটাবেজ রিসোর্সের সর্বাধিক ব্যবহার নিশ্চিত করা যায়।
 ঘ) নেটওয়ার্ক তৈরি করতে তুলনামূলক খরচ কর লাগে।

- অসুবিধা:
- ইহা পরিচালনার জন্য জটিল সফটওয়্যার এর প্রয়োজন হয়।
 - ব্যক্তিগত সিস্টেম তুলনামূলক দুর্বল।
 - কম্পিউটার সমূহের মধ্যে নেটওর্ক এর জন্য উচ্চগতি সম্পর্ক নেটওর্ক কিংবা টেলিফোন লাইন এর প্রয়োজন হয়।
 - Centralized System এর তুলনায় ডাটা নিরাপত্তা কম।

প্রশ্ন 6. Distributed Database System এর ব্যবহার লিখ।
উত্তর: Distributed Database System এর ব্যবহার নিম্নরূপ:

- অর্থনৈতিক ক্ষেত্রে।
- প্রাতিশালিক ক্ষেত্রে।
- চিকিৎসা ক্ষেত্রে।
- শিক্ষা ক্ষেত্রে।
- ইলেক্ট্রনিক ব্যবসার ক্ষেত্রে।
- ব্যাংক সেবার এ।
- ইনস্যুলেন অফিস এ।
- ইনভেন্টরী Control System এ।
- পরিসংখ্যান বিভাগে।
- স্টক বিশ্বেষণ ইত্যাদি ক্ষেত্রে।
- তথ্য যোগাযোগ ক্ষেত্রে।
- ডাটা সংযোগে।
- রিজার্ভেশন সিস্টেমে।
- শিল্প ক্ষেত্রে।
- Information sharing ইত্যাদি ক্ষেত্রে।

প্রশ্ন 7. Distributed Database system এর স্থানান্তরণে অবকাঠামো অনেকন করুন?



প্রশ্ন 8. Distributed Database system এ ব্যবহৃত বিভিন্ন ধরনের Topology এর নাম লিখ?

উত্তর: Distributed Database system এ ব্যবহৃত বিভিন্ন ধরনের Topology নাম নিম্নরূপ:

- Fully connected Topology
- Partially connected Topology
- Tree Topology
- Star Topology
- Ring

প্রশ্ন 9. নো এসকিউএল ডেটাবেজ কি? উদাহরণ দাও?

উত্তর: নো এসকিউএল ডেটাবেজ (No-SQL database) : NoSQL stands for "not only SQL". NoSQL হল database management এর একটি পদ্ধতি। যা key-value, document, columnar and graph formats সহ বিভিন্ন ধরণের ডেটা মডেলকে সমর্থন করতে পারে। এটি traditional relational databases এর বিকল্প এবং large set of distributed data এর জন্য বেশি উপযোগী।

Examples of NoSQL database:

- MongoDB, CouchDB, Cloudant
- Memcached, Redis, Coherence (key-value store)
- HBase, Big Table, Accumulo (Tabular)

প্রশ্ন 10. ক্লাউড ডেটাবেজ কি? উদাহরণ সহ সুবিধা লিখ?

উত্তর: ক্লাউড ডেটাবেজ (Cloud database) : Cloud Database হল এক ধরনের database service যাহা cloud platform এ সংরক্ষিত এবং online এর মাধ্যমে access করা যাব। Cloud databases কে Database as Service (DBaaS) ও কলা হয় কারণ তারা managed service প্রদান করে। Some best cloud options are:

- AWS (Amazon Web Services)
- Snowflake Computing
- Oracle Database Cloud Services
- Microsoft SQL server
- Google cloud spanner

ক্লাউড ডাটাবেসের সুবিধা:

Lower costs- যেহেতু একাধিক ডাটা সেন্টার থেকে maintain এ support দেয়া হয় তাই সাধারণত company গুলোকে invest করাতে হয়ন।

Automated - ক্লাউড ডেটাবেসগুলি বিভিন্ন স্বত্ত্বান্তর প্রক্রিয়া দ্বারা সম্ভূত। বেমনঃ recovery, failover, and auto-scaling.

Increased accessibility - যে কোন জায়গা থেকে যে কোন সমৃদ্ধ ক্লাউড-ভিত্তির ডাটাবেস অ্যাক্সেস করা যায়। শুধুই ইন্টারনেট সংযোগ প্রয়োজন।

প্রশ্ন 11. অবজেক্ট অরিয়েন্টেড ডেটাবেজ কি?

উত্তর: অবজেক্ট অরিয়েন্টেড ডেটাবেজ: Object-Oriented Database হল এক ধরনের database management system, যা Object-Oriented Programming (OOP) ব্যবহার করে data কে object এ উপস্থাপন করে। একে object database ও কলা হয়। এটি relational database (table-oriented) থেকে তিনি।

প্রশ্ন 12. অবজেক্ট অরিয়েন্টেড ডেটাবেজ ম্যানেজমেন্ট সিস্টেম কি?

[What is Object Oriented Database Management System]

উত্তর: (OODBMS): Object oriented database management system (OODBMS) কে Object database management system (ODBMS) ও কলা হয়। এটি database এবং object oriented programming (OOP) language সমরয়ে গঠিত। এটি programmer দের কে product develop করার অনুমতি প্রদান করে, object হিসেবে সংরক্ষণ করে, এবং

বিশ্বাস কে নতুন object এ রূপান্তর করে। object oriented programming language এর সাথে database যুক্ত করা কারণে, এটি একটি environment এর মধ্যে consistency রক্ষণ করে।

বিশ্বাস এর মধ্যে এমনভাবে তৈরি করা হয়েছে যেন object oriented programming language (বেমন- Delhi, Ruby, Python, JavaScript, Perl, C#, Visual Basic, .NET, C++, Objective-C ইত্যাদি) এর সাথে দক্ষতার সাথে কাজ করতে সক্ষম।

প্রশ্ন 13. What are the ways for no data loss, describe? [কোন ডাটা না হারানোর উপায় কি, বর্ণনা করুন?]

উত্তর: যখন information system গুলি corrupted, যুক্ত ফেলা বা পুনরুদ্ধার করা যায় না তখন ডেটা loss হয়। এটি সাধারণত স্টোরেজ, ট্রান্সিশন বা প্রক্রিয়াকরণে অবহেলার কারণে ঘটে।

The ways for no data loss

- Always back up your data
- Diversify your backups
- Encrypt sensitive data
- Address data security
- Use anti-virus and email security
- Trust the professionals
- Firewall and Antivirus
- Protect Data from Power Surges
- Develop a Disaster Recovery Plan
- Keep Your Computer Dust-Free and Dry
- Specify Access Levels
- Work With IT Security Experts

॥ Data Model ॥

প্রশ্ন 1. Data model বলতে কি? ডাটা মডেলের সুবিধা-অসুবিধা সমূহ লিখ। [Write the advantages and disadvantages of the data model?]**

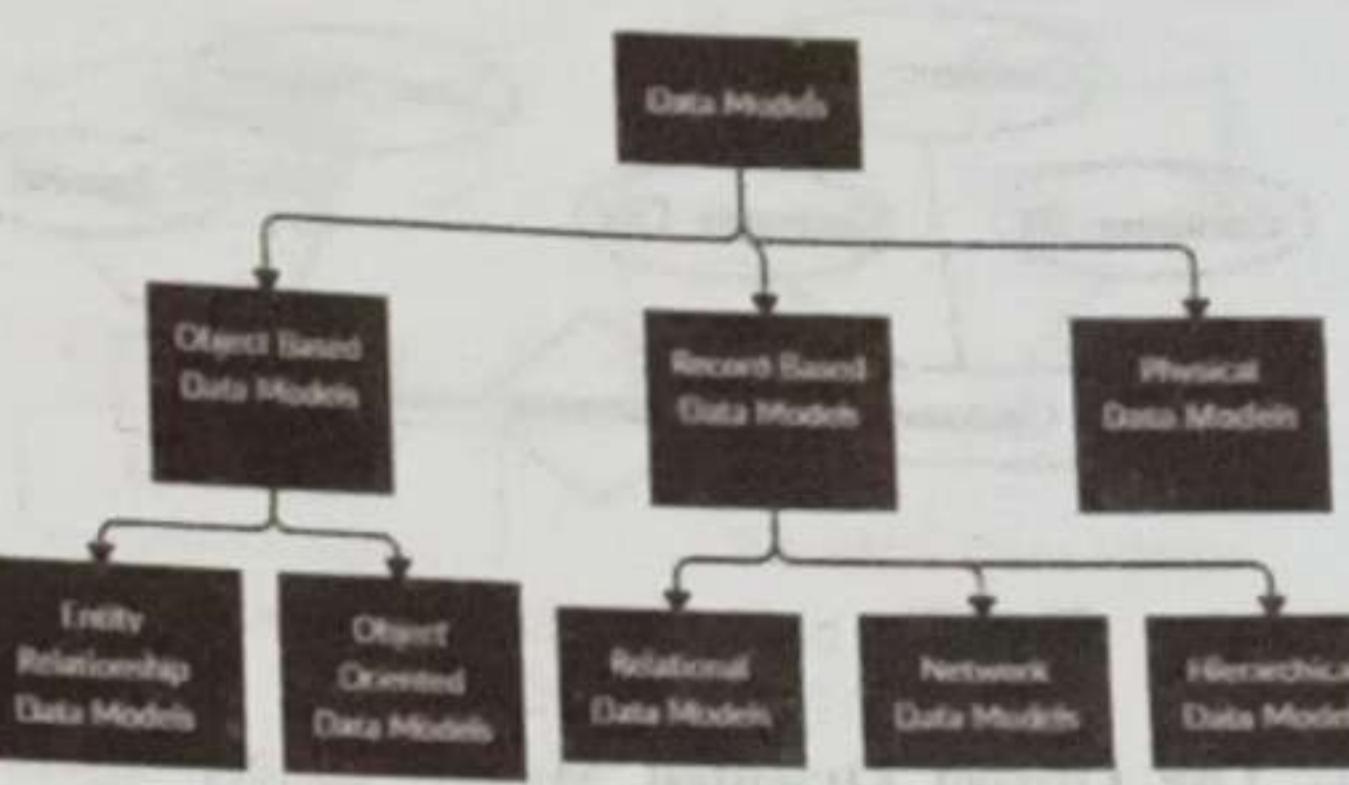
উত্তর: Logical & View Level এ ডাটা বর্ণনা করার জন্য যে ডাটা মডেল ব্যবহার করা হয় তাকে Object based logical model বলা হয়।

Object based logical model দুই অংশে বিভক্ত যথাঃ-

ক) Entity Relationship Model : কতগুলো Basic Object এবং তাদের মধ্যে পরিশ্বারিক সম্পর্ক যে ডাটা মডেল এর মাধ্যমে Graphically উপস্থাপন করা হয়, তাকে E-R মডেল বা The entity relationship model বলে।

উদাহরণঃ ধরি, Customer এবং loan দুটি এনটিটি সেট এবং তাদের

মধ্যে Borrower হল রিলেশনশীপ সেট, কাউন্টিমার এবং loan সেট এর একটি বিভিন্ন গুলোকে Borrower রিলেশনশীপ এর মাধ্যমে সম্পর্ক স্থাপন করে দেখানো হলো :



Advantage of a data model:

1. Increased effectiveness
2. Reduced costs
3. simplicity
4. Minimum redundancy
5. data integrity
6. In data independence
7. faster performance
8. Reduced errors

Disadvantages of Data Model:

- Complexity
- Size
- Cost of DBMS
- Cost of conversion
- Performance
- Higher impact of a failure etc.

প্রশ্ন 2. Object Based Logical Model এর সংক্ষিপ্ত বর্ণনা দাও?

উত্তর: Logical & View Level এ ডাটা বর্ণনা করার জন্য যে ডাটা মডেল ব্যবহার করা হয় তাকে Object based logical model বলা হয়।

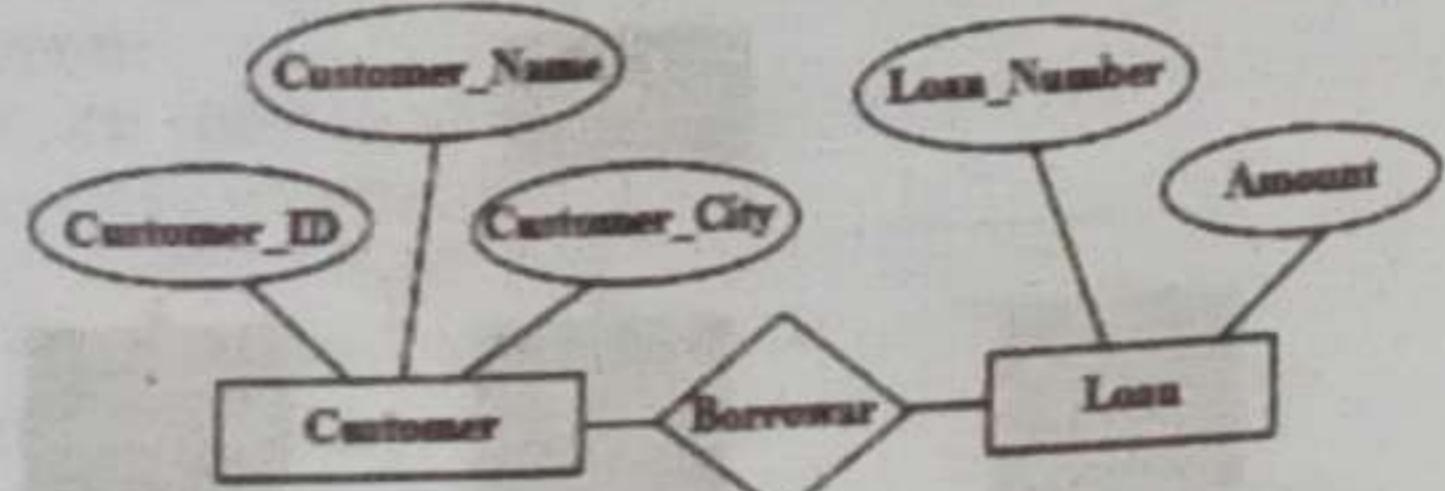
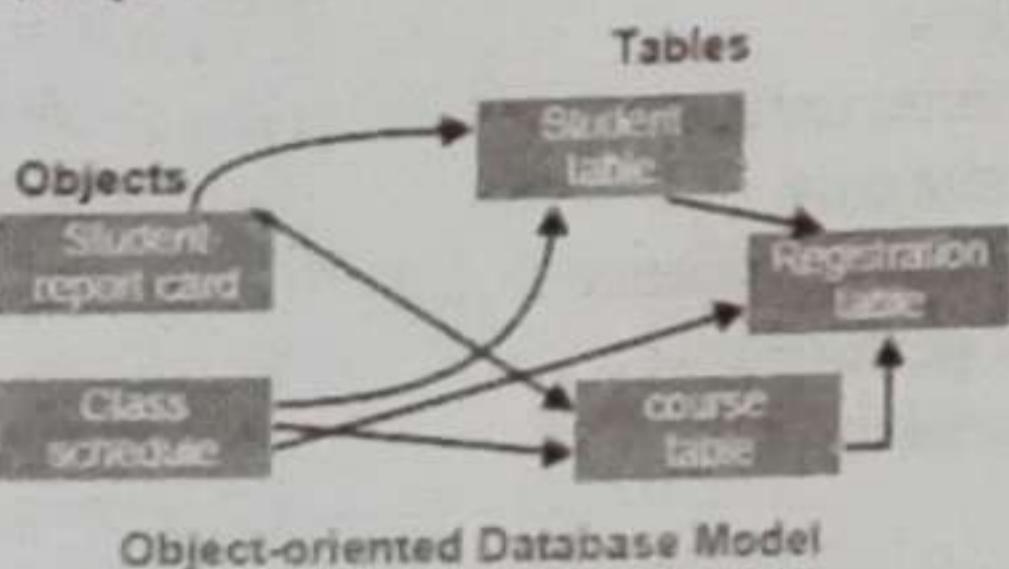


Fig: The entity relationship model

৪) **The Object Oriented Model:** Object Oriented Model কতগুলো Object এর উপর ভিত্তি করে প্রতিষ্ঠিত। একেতে একটি অবজেক্টে তার Instance Variable এ জমাকৃত ডাটার মান ধারণ করে থাকে। যদি কতকগুলো অবজেক্টে একই ধরনের Value ধারণ করে তখন তাকে ক্লাস বলে।



প্রশ্ন 3. Record Based logical models তলোর সংক্ষিপ্ত বর্ণনা দাও।

উত্তরঃ ডাটা মডেল এর অর্থন্ত Record based logical model সমূহকে logical এবং View level এ বর্ণনা করা হয়। Object based logical model এর সাথে এর মূল পার্শ্বক্য হলো, এখানে ডাটাবেসের সাথে সম্পর্ক কাঠামো সুস্পষ্ট।

Record Based Logical Models 3 টি অশে বিভক্ত।

ক) **Relational Data Model:** যে ডাটা মডেল এ ডাটা এবং তাদের রিলেশনশিপ কে টেবিল এর মাধ্যমে উপস্থাপন করা হয় তাকে Relational Data Model বলে।

Operation:

- a) Select
- b) Union
- c) Project
- d) Set difference
- e) Cartesian Product
- f) Rename

নিম্নে এর একটি চিত্র দেখানো হলোঃ

Stdname	Std Id	City	Dept.	CGPA
Jamal	174024	Feni	CSE	3.85
Rakib	174026	Khulna	ME	3.75
Harun	174025	New York	EEE	3.83

Fig: Relational data model.

৫) **Network Data model:** যে ডাটা মডেল এ ডাটা কে Collection of Record এবং তাদের মধ্যে রিলেশনশিপ কে যদি Link দ্বারা প্রকাশ করা হয়, তবে তাকে নেটওর্ক ডাটা মডেল বলে। নিম্নে এর একটি চিত্র দেখানো হলোঃ

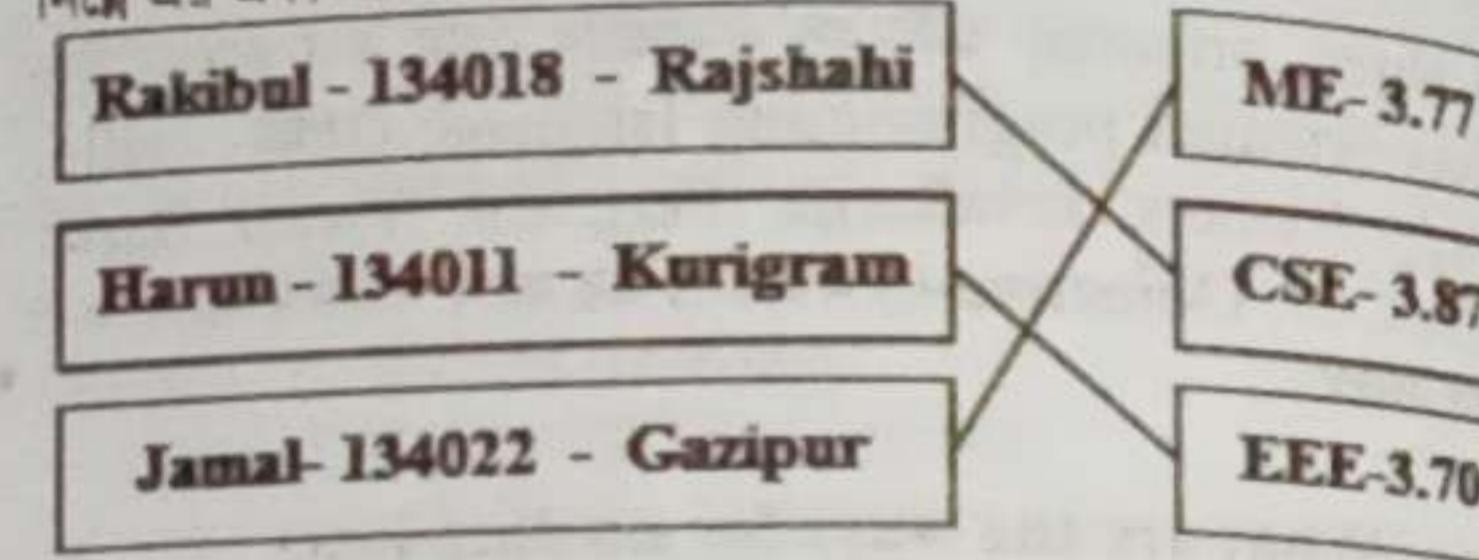
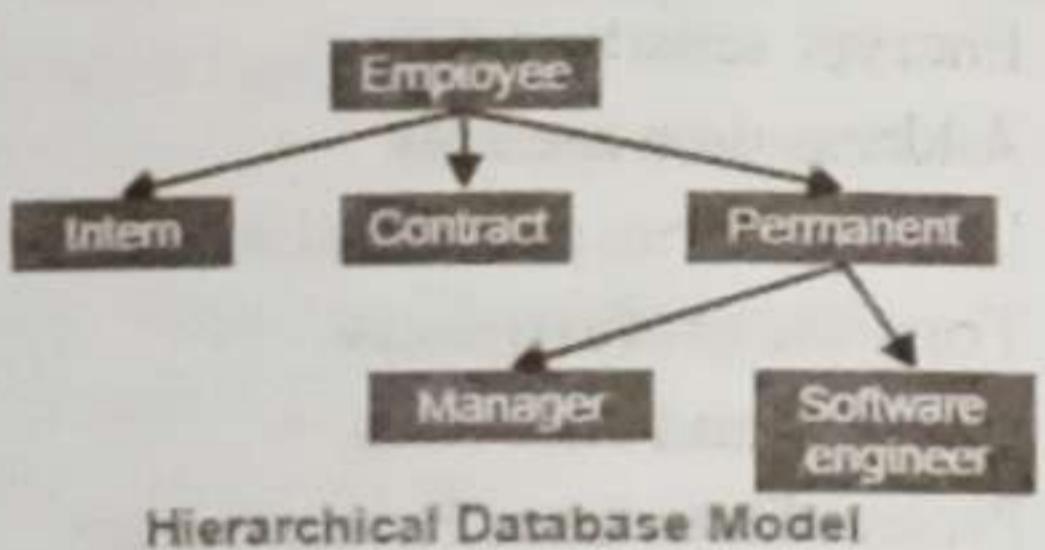


Fig: Network Data Model.

৬) **Hierarchical Data Model:** Record based logical model এর অর্থন্ত যে ডাটা মডেল এ ডাটা এবং এদের মধ্যকার রিলেশনশিপ কে Link দ্বারা Tree আকারে প্রকাশ করা হয়, তাকে Hierarchical Data Model বলে। নিম্নে এর একটি চিত্র দেখানো হলোঃ



Hierarchical Database Model

প্রশ্ন 4. রিলেশনাল ডাটা মডেল এর কাজ লিখ। [Write the function of relational data model?] [BNPS-2014,Dept of ICT-2014]

উত্তরঃ

RDBMS এর Function (কাজ):

- ডেটাতে সহজ আ্যোগ্য।
- ডেটা সুরক্ষার জন্য আরও ভাল নিরাপত্তা।
- সব ধরনের সম্পর্কের প্রতিনিধিত্ব।
- ডেটা অপ্রযোজ্যনীয়তা হাস করা হয়।
- প্রযোজনীয় ডেটার দ্রুত এবং দক্ষ অনুসৃতান।
- সহজে পরিবর্তন এবং ডেটা মুছে ফেলা।
- ডেটার সমসাময়িক আ্যোগ্য।
- ডেটা আ্যোগ্য করার জন্য ইউজার ইন্টারাফেস প্রদান করা।
- মান প্রযোগ করে সততা বজায় রাখা।

প্রশ্ন 5. Network Model & Hierarchical Data Model মধ্যে পার্শ্বক্য (Distinguish) লিখ?**

উত্তরঃ Network Model & Hierarchical Data Model মধ্যে পার্শ্বক্য নিম্নরূপঃ

Network ডাটা মডেল	Hierarchical ডাটা মডেল
১। Record Based logical model এর অর্থন্ত যে ডাটা মডেল এ ডাটা কে Collection of Record	১। Record based logical model এর অর্থন্ত যে ডাটা মডেল এ ডাটা এবং এদের মধ্যকার রিলেশনশিপ কে

এবং তাদের মধ্যে রিলেশনশিপ কে যদি Link দ্বারা প্রকাশ করা হয়, তবে তাকে Hierarchical Data Model বলে।

- ২। এতে রেকর্ড সমূহ পাশাপাশি অবস্থান করে।
- ৩। একে ডাটা স্ট্রাকচার ডায়াগ্রাম দ্বারা প্রকাশ করা হয়।
- ৪। Many to One Support করে।
- ৫। এতে Dummy Node থাকে না।
- ৬। একে Tree স্ট্রাকচার ডায়াগ্রাম দ্বারা প্রকাশ করা হয়।
- ৭। Many to One Support করে না।
- ৮। এতে Dummy Node থাকে।

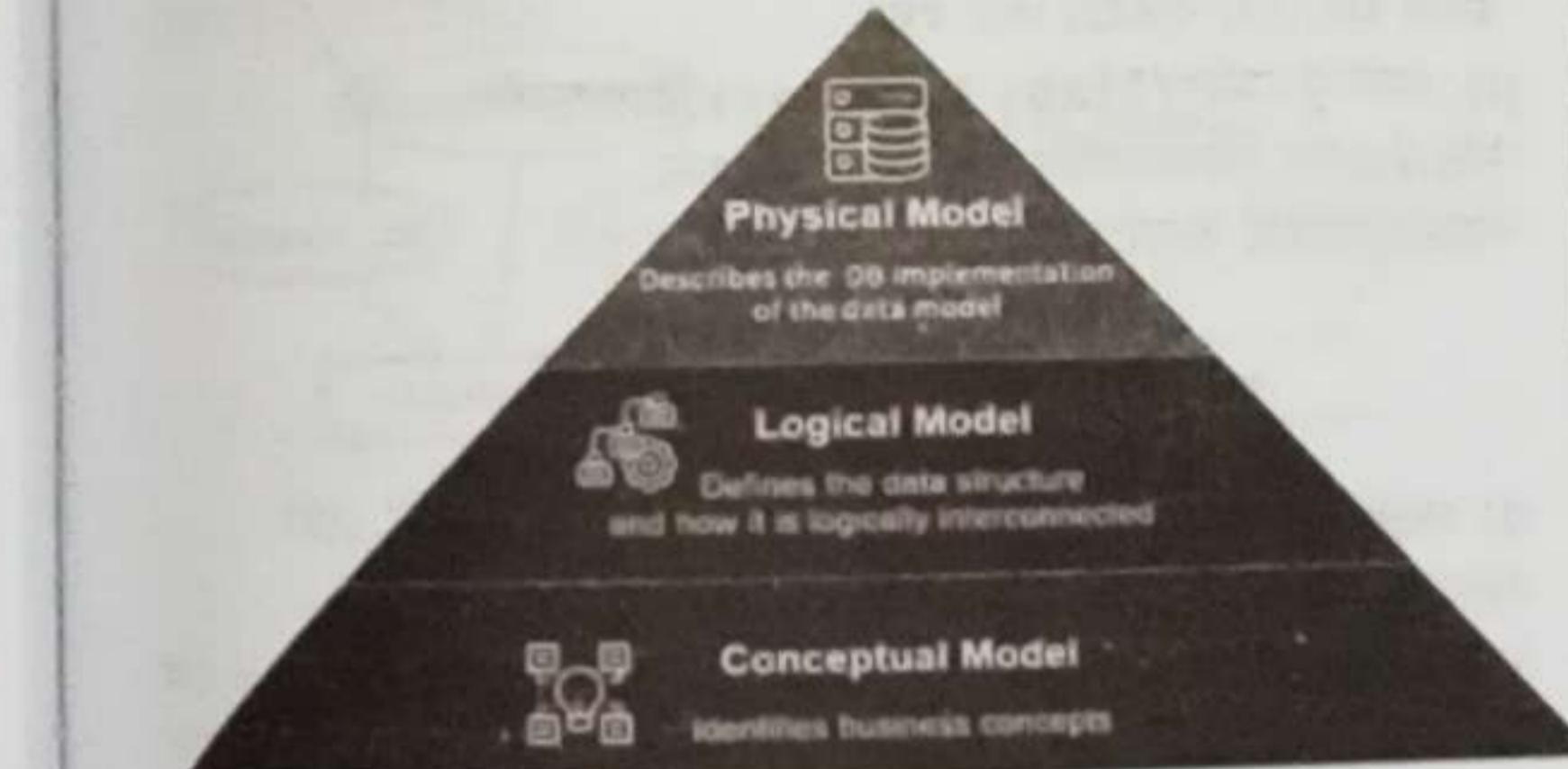
প্রশ্ন 6. ডাটা মডেল কেন ব্যবহার করা হয় ? Why data models are used?

উত্তরঃ

Data model এর ব্যবহার:

- a) Data structure বর্ণনা করার জন্য।
- b) Data manipulation বর্ণনা করার জন্য।
- c) Data integrity বর্ণনা করার জন্য।
- d) Data relationship বর্ণনা করার জন্য।
- e) Data semantics বর্ণনা করার জন্য।
- f) Consistency constraints বর্ণনা করার জন্য।

প্রশ্ন 7. ডাটা মডেলিং এর ধাপ কয়তি ও কি কি? What are the steps of data modeling?**



উত্তরঃ ডাটা মডেল ও প্রকার : যথা

Conceptual ডাটা মডেল : যে ডাটা মডেল এ ডাটাবেস সিস্টেম সম্পর্কে পূর্ব পরিকল্পনা এবং তাদের রিলেশন সম্পর্কে Represent করে থাকে, তাকে Conceptual ডাটা মডেল বলে। Semantic Data model মূলত একটি Abstract Data model যা জমাকৃত simple ডাটা কে Real world এর সাথে সম্পর্ক তৈরীতে, সাহায্য করে থাকে। Semantic Data Model কে Conceptual Data Model ও বলা হয়।

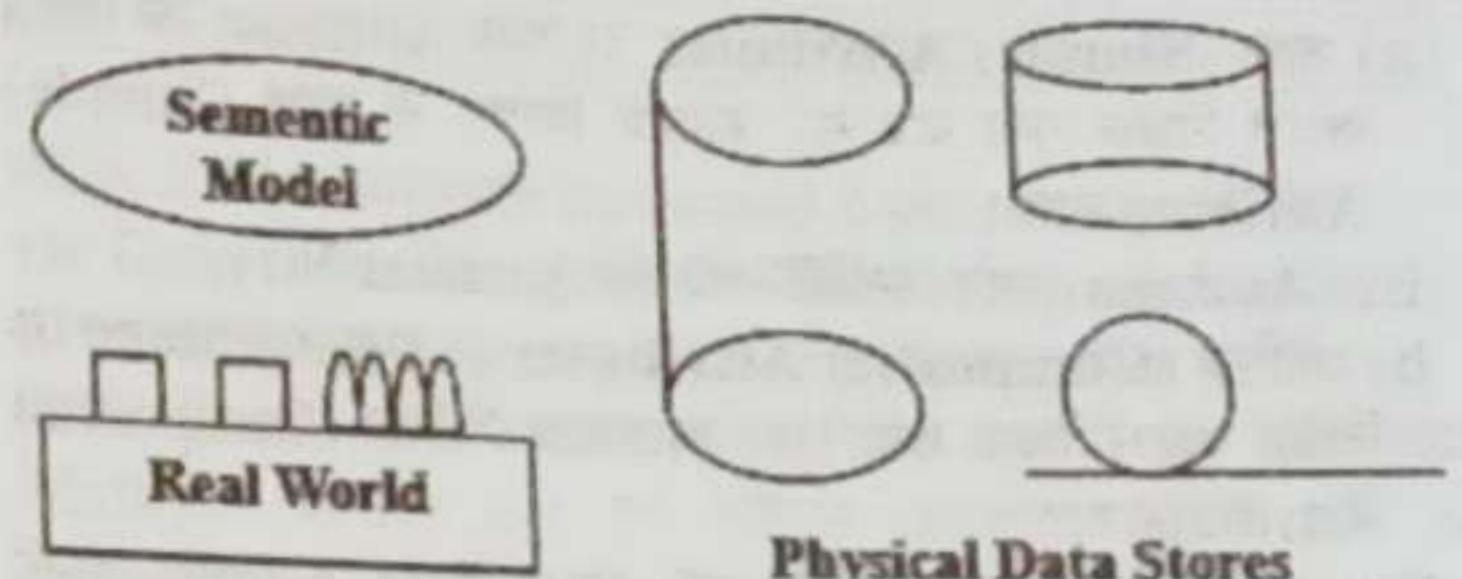


Fig: Semantic Data Stores.

Logical ডাটা মডেল : ডিবিএমএস নির্বিশেষ সিস্টেমটি কীভাবে প্রয়োগ করা উচিত তা নির্ধারণ করে। এই মডেলটি সাধারণত ডেটা আর্কিটেকচার এবং ডেটা স্ট্রাকচারের সময়ে গঠিত মডেল।

Physical ডাটা মডেল : একটি physical data model হল database-specific model যা রিলেশনাল ডেটা অবজেক্ট (উদাহরণস্বরূপ, টেবিল, কলাম, primary and foreign key) এবং তাদের সম্পর্ককে represent করে। একটি physical data মডেল DDL statement তৈরি করতে ব্যবহার করা হয় যা পরে একটি ডাটাবেস সার্ভারে স্থাপন করা হয়।

প্রশ্ন 8. Instance কাকে বলে? Or Database এর Instance কি? উভয় কোন একটি নির্দিষ্ট ঘূর্ঘন্তে ডাটাবেস এ জমাকৃত ডাটা। ইনফরমেশন এর সমষ্টিকে Instance বা Database Instance বলে।

E-R Diagram

প্রশ্ন 1. E-R Diagram কাকে বলে ? [PSC-2017]***

উত্তরঃ Entity relationship model এ Attribute এর Entity Set এবং Entity Set এর সাথে relationship যে ডায়াগ্রাম এর মাধ্যমে দেখানো হয়, তাকে E-R Diagram বলে। Relational database design করতে E-R Diagram ব্যবহার করা হয়।

প্রশ্ন 2. Entity কি? entity set কলতে কি বুঝা? [MULTIPLE MINISTRY-2016]

উত্তরঃ এনটিটি হল একই রকম কতগুলো অবজেক্ট এর একটি গ্রুপ। যারা একই ধরনের প্রোপার্টি বা Attribute শেয়ার করে থাকে। যেমনঃ

Entry ——————> Name Dept Roll No CGPA ——————> Entry Set

এখানে Name, Dept, Roll, CGPA সবগুলোকে একত্রে এনটিটি সেট বলে। এবং প্রত্যক্ষটিকে এর একটি এনটিটি বলে। যেমনঃ Name, Dept।

প্রশ্ন 3. Attribute কি? এর শ্রেণী বিভাগগুলো উল্লেখ কর এবং সংক্ষেপে বর্ণন লিখ। Mention its class sections and write a brief description. [Multiple Ministry-2016]***

উত্তরঃ Attribute: কোন অবজেক্ট এর একক বৈশিষ্ট্যকে এই অবজেক্ট এর Attribute বলে। E-R Diagram এ ব্যবহৃত বিভিন্ন ধরনের Attribute সমূহকে নিম্নলিখিত শ্রেণীতে বিভক্ত করা যায়। যথা:

a) **সরল (Simple) Attribute:** যে সমস্ত Attribute কে বিভিন্ন অংশে বিভক্ত করা যায় না, তাকেই সিম্পল বা সরল (Simple) Attribute বলে।

Ex: Account নামক এনটিটি সেট এর Account No।

b) **মৌলিক (Composite) Attribute:** যে সমস্ত Attribute কে বিভিন্ন অংশে বিভক্ত করা যায়, তাদেরকে মৌলিক (Composite) Attribute বলে।

Ex: কাস্টমার নামক এনটিটি সেট এর কাস্টমার Name, কারন First name, Middle name ও Last name ইত্যাদি তাদের ভাগ করা যায়।

c) **একক (Single Valued) Attribute:** যে সমস্ত Attribute কে অধুনার একটি মান দ্বারা নির্দিষ্ট করা যায়, তাদেরকে একক (Single Valued) Attribute বলে।

Ex: Lone নামক এনটিটি সেট এর Lone নাম।

d) **বহুমান (Multivalued) Attribute:** যে সমস্ত Attribute এর অনেকগুলো মান থাকতে পারে, তাদেরকে বহুমান (Multivalued) Attribute বলে।

e) **Null Attribute:** যদি কোন Attribute এর কোন মান না থাকে, তবে তাকে Null Attribute তাকে বলে।

f) **Dervied Attribute:** যদি কোন Attribute এর মান অন্য কোন Attribute হতে পাওয়া যায়, তবে এ Attribute কে Dervied Attribute বলে।

প্রশ্ন 8. E-R Diagram এর বিভিন্ন সিম্বল এর কাজ শিখ।***

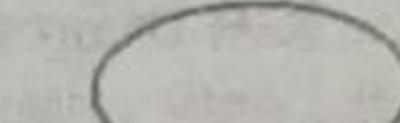
উত্তর: E-R Diagram এর বিভিন্ন সিম্বল এর কাজ নিচেরপৰি-

a) **আয়তক্ষেত্র (Rectangle):**



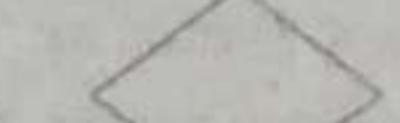
করার জন্য Single Rectangle ব্যবহার করা হয়।

b) **উপবৃত্ত (Ellipses):**



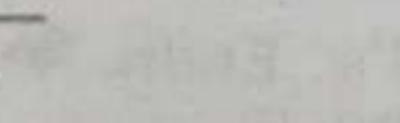
Entity Set এর Attribute কে প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

c) **ত্যাগক্ষণ (Diamond):**



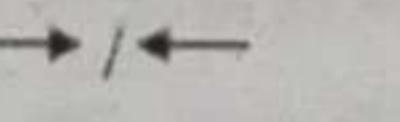
বিলেশনশীল সেট কে প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

d) **লাইন (Line):**



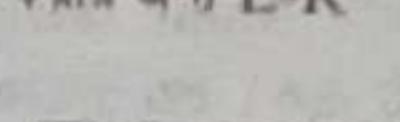
লাইন দ্বারা একটিভিউট কে এন্টার সেট এর সাথে এবং এনটিটি সেট কে বিলেশনশীল সেট এর সাথে লিঙ্ক করা হয়।

e) **ডাবল আয়তক্ষেত্র (Double rectangle):**



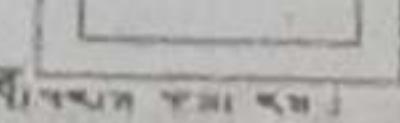
Weak Entity set কে প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

f) **ডাবল উপবৃত্ত (Double Ellipse):**



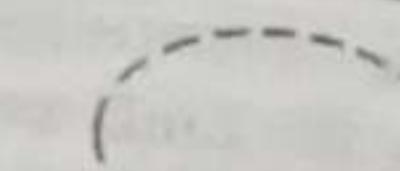
Multivalued Attribute কে প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

g) **ডাবল ত্যাগক্ষণ (Double Diamond):**

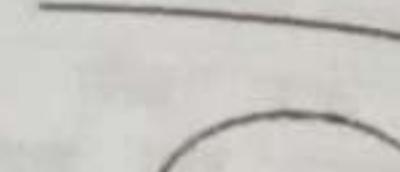


Weak Entity Set এর সাথে বিলেশনশীল সেট কে প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

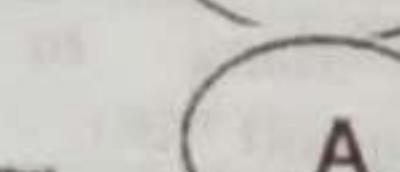
h) **ডাল উপবৃত্ত (Dashed Ellipse):**
Derived Attribute কে প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।



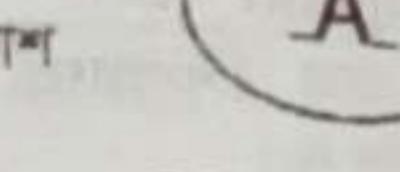
i) **ডাবল লাইন (Double Line):**
বিলেশনশীল সেট এর মধ্যে একটি এনটিটি সেট এর টোটাল Participation কে নির্দেশ করার জন্য ইহা ব্যবহার করা হয়।



j) **প্রাইমারী কী (Primary Key):**
প্রাইমারী কী কে উপহাসন করার জন্য ইহা ব্যবহার করা হয়।



k) **উইক এনটিটি সেট এর ডিসক্রিমিনেটিং এনটিটি সেট এর প্রকাশ**



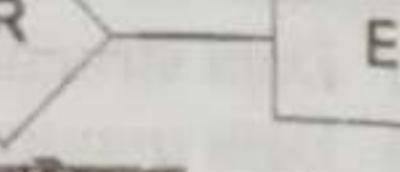
উইক এনটিটি সেট এর ডিসক্রিমিনেটিং এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

l) **টোটাল পার্টিসিপেশন এনটিটি সেট এর প্রকাশ**



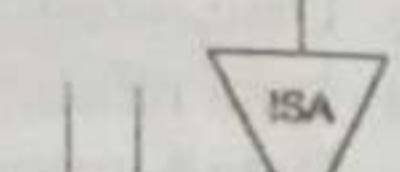
টোটাল পার্টিসিপেশন এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

m) **আই এস এস (আইডি এন্টিটি এনটিটি সেট এর প্রকাশ)**



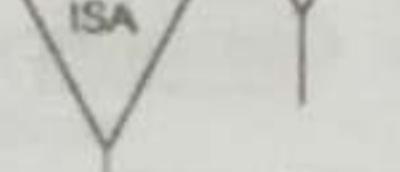
আই এস এস (আইডি এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

n) **টোটাল জেনারেলাইজেশন এনটিটি সেট এর প্রকাশ**



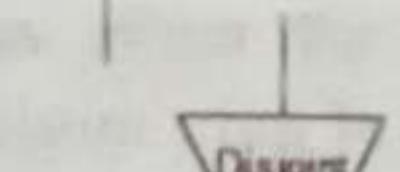
টোটাল জেনারেলাইজেশন এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

o) **ডিজন্যুটে জেনারেলাইজেশন এনটিটি সেট এর প্রকাশ**



ডিজন্যুটে জেনারেলাইজেশন এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

p) **মেনি-টু-মেনি (Many to Many) রিলেশনশীল এন্টিটি এনটিটি সেট এর প্রকাশ**



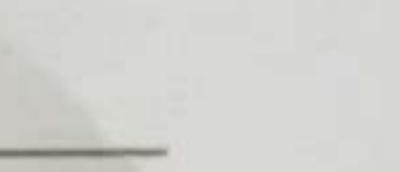
মেনি-টু-মেনি রিলেশনশীল এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

q) **ওয়ান টু ওয়ান (One to One) রিলেশনশীল এন্টিটি এনটিটি সেট এনটিটি সেট এর প্রকাশ**



ওয়ান টু ওয়ান রিলেশনশীল এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

r) **মেনি টু ওয়ান রিলেশনশীল এন্টিটি এনটিটি সেট এর প্রকাশ**



মেনি টু ওয়ান রিলেশনশীল এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

s) **লাইন এন্টিটি এনটিটি সেট এর প্রকাশ**



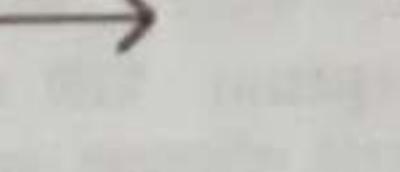
লাইন এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

t) **ডাবল লাইন এন্টিটি এনটিটি সেট এর প্রকাশ**



ডাবল লাইন এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

u) **ডাবল ডায়াম্যান্ড এন্টিটি এনটিটি সেট এর প্রকাশ**



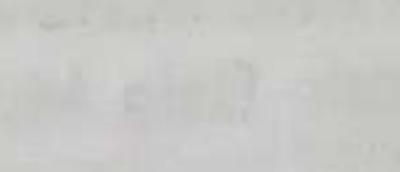
ডাবল ডায়াম্যান্ড এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

v) **ডাল এন্টিটি এনটিটি সেট এর প্রকাশ**



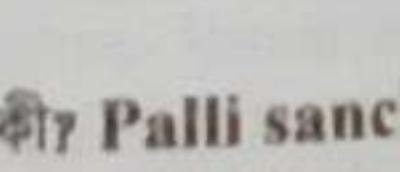
ডাল এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

w) **টোটাল পার্টিসিপেশন এন্টিটি এনটিটি সেট এর প্রকাশ**



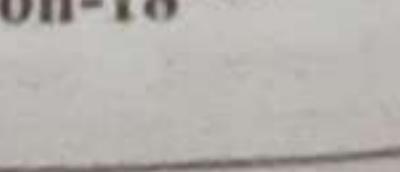
টোটাল পার্টিসিপেশন এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

x) **আইডি এন্টিটি এনটিটি সেট এর প্রকাশ**



আইডি এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

y) **জেনারেলাইজেশন এন্টিটি এনটিটি সেট এর প্রকাশ**



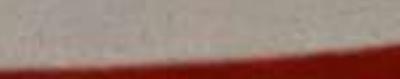
জেনারেলাইজেশন এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

z) **ডিজন্যুটে জেনারেলাইজেশন এন্টিটি এনটিটি সেট এর প্রকাশ**



ডিজন্যুটে জেনারেলাইজেশন এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

aa) **ডাবল ডায়াম্যান্ড এন্টিটি এনটিটি সেট এর প্রকাশ**



ডাবল ডায়াম্যান্ড এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

bb) **ডাল এন্টিটি এনটিটি সেট এর প্রকাশ**

ডাল এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

cc) **টোটাল পার্টিসিপেশন এন্টিটি এনটিটি সেট এর প্রকাশ**

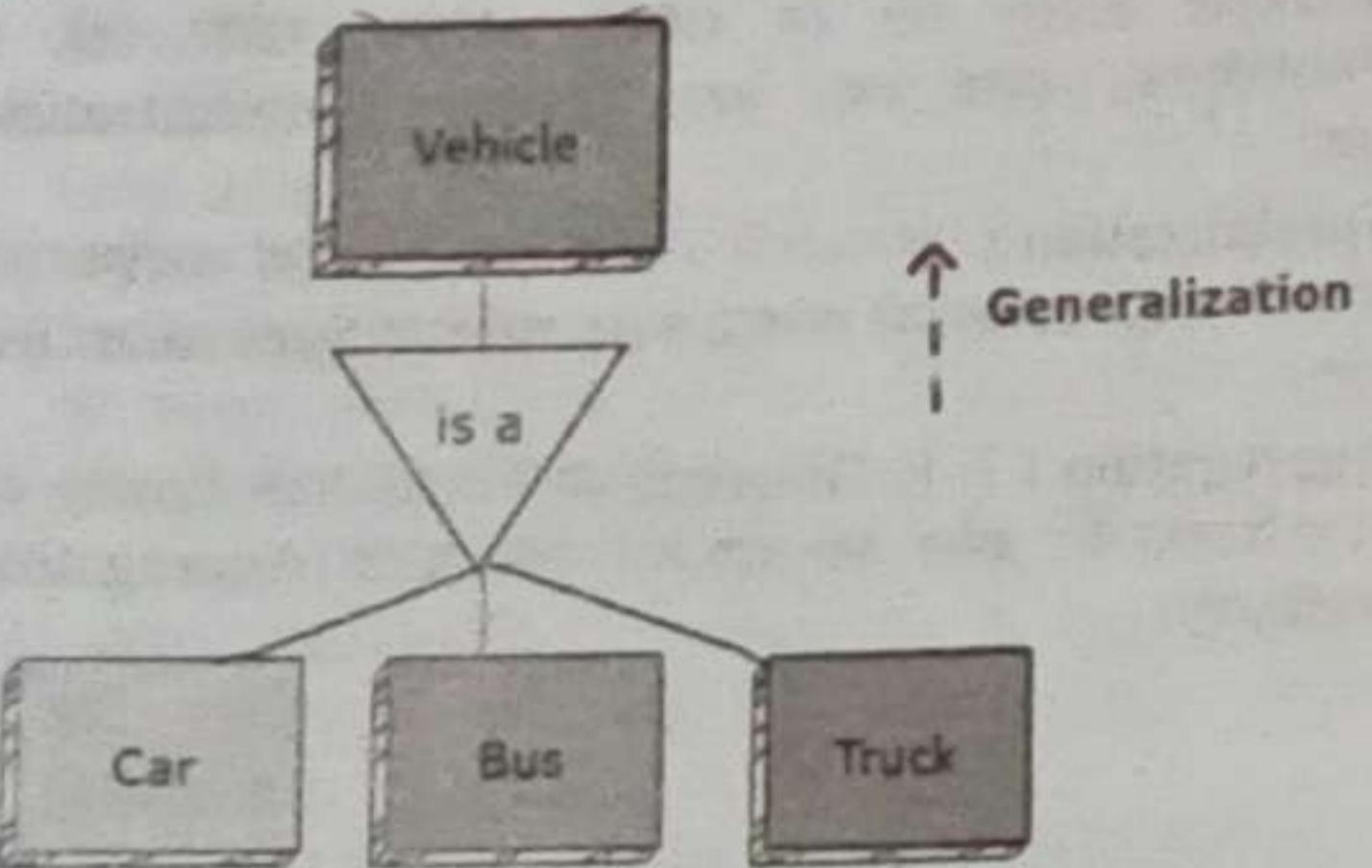
টোটাল পার্টিসিপেশন এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

dd) **আইডি এন্টিটি এনটিটি সেট এর প্রকাশ**

আইডি এন্টিটি এনটিটি সেট এর প্রকাশ করার জন্য ইহা ব্যবহার করা হয়।

specialization (বিশেষাকরণের) বিপরীত হিসাবেও তাৰা ঘৰে পাৰে।

নিম্নলিখিত enhanced entity relationship চিত্ৰটি generalization (সাধাৰণীকৰণ) প্ৰদৰ্শনেৰ জন্য একটি hierarchical database (শ্ৰেণিবিক ডাটাবেস) entity কে প্ৰকাশ কৰে:



প্ৰশ্ন ১০. তাৰা স্থোকচাৰ ভাষ্যাম কি? What is a data structure diagram?

উভয়ৰ ভাষ্যাম স্থোকচাৰ diagram এমন একটি diagram যা entity এবং তাৰের relationship কে representation কৰে। এৰ Basic দুটি কলোনেট রয়েছে।

ক) Box [] যা Record type কে Represent কৰে।
খ) Lines ————— যা Record type এৰ মধ্যে Link কে Represent কৰে।

প্ৰশ্ন ১১. এনটিটি রিলেশনশীপ কি? [Multiple Ministry-2016]***
উভয়ৰ বিভিন্ন প্ৰকাৰ এনটিটি সমূহৰ মধ্যে সম্পর্ক ছাপন কৰাই রিলেশনশীপ। অন্যাবে বলা যায় বিভিন্ন প্ৰকাৰ এনটিটি সমূহৰ মধ্যে Aossoication হচ্ছে রিলেশনশীপ।

প্ৰশ্ন ১২. ডেটাবেজ রিলেশন কি? ডেটাবেজ রিলেশনেৰ শৰ্ত এবং ডেটাবেজ রিলেশন প্ৰকাৰতে লিখুন? [What is a database relation? Write the conditions of the database relation and the types of database relation]

উভয়ৰ ডেটাবেজ রিলেশন - ডেটাবেজৰ একটি টেবিলৰ রেকৰ্ডেৰ সাথে অন্য এক বা একাধিক টেবিলৰ রেকৰ্ডেৰ সম্পর্ককে ডেটাবেজ রিলেশন বলে। অৰ্থাৎ ডেটাবেজ রিলেশন হলো বিভিন্ন ডেটা টেবিলৰ মধ্যকাৰ সম্পৰ্ক।

রিলেশনেৰ শৰ্তসমূহ:

১. রিলেশনাল ডেটা টেবিলগুলোৰ মধ্যে কমপক্ষে একটি কমন ফিল্ড থাকবে। কমন ফিল্ডেৰ ডেটা টাইপ, ফিল্ড সাইজ এবং ফরমেট ইত্যাদি একই হতে হবে।
২. রিলেশনাল টেবিলগুলোৰ মধ্যে অস্তত একটি টেবিলে অবশ্যই প্ৰাইমেৰি কি ফিল্ড থাকতে হবে।

রিলেশনেৰ প্ৰকাৰতেস:

১. One to One রিলেশন।

২. One to Many রিলেশন।
৩. Many to One রিলেশন।
৪. Many to Many রিলেশন।

প্ৰশ্ন ১৩. ম্যাপিং কাৰতিনালিটিজ (Mapping Cardinality) কৰতে কি বুঝ? [BB-2016] **

উভয়ৰ DBMS এ কোন এনটিটি সেটেৰ একটি এনটিটি অপৰ একটি এনটিটি সেটেৰ কোনো এনটিটিৰ সাথে এসোসিয়েশন সৃষ্টি কৰতে পাৰে তা যে প্ৰকাশ কৰা হয় তাকে ম্যাপিং কাৰতিনালিটি বলে। এটি বাইনাৰি রিলেশনশীপ এৰ ক্ষেত্ৰে ব্যবহৃত হয়।

প্ৰশ্ন ১৪. ম্যাপিং কাৰতিনালিটিজ কত প্ৰকাৰ? সংক্ষিপ্ত বৰ্ণনা দাও। How many types of mapping cardinality? Give a brief description.

উভয়ৰ ম্যাপিং কাৰতিনালিটি ৪ প্ৰকাৰ। যথাঃ

a) One to one:

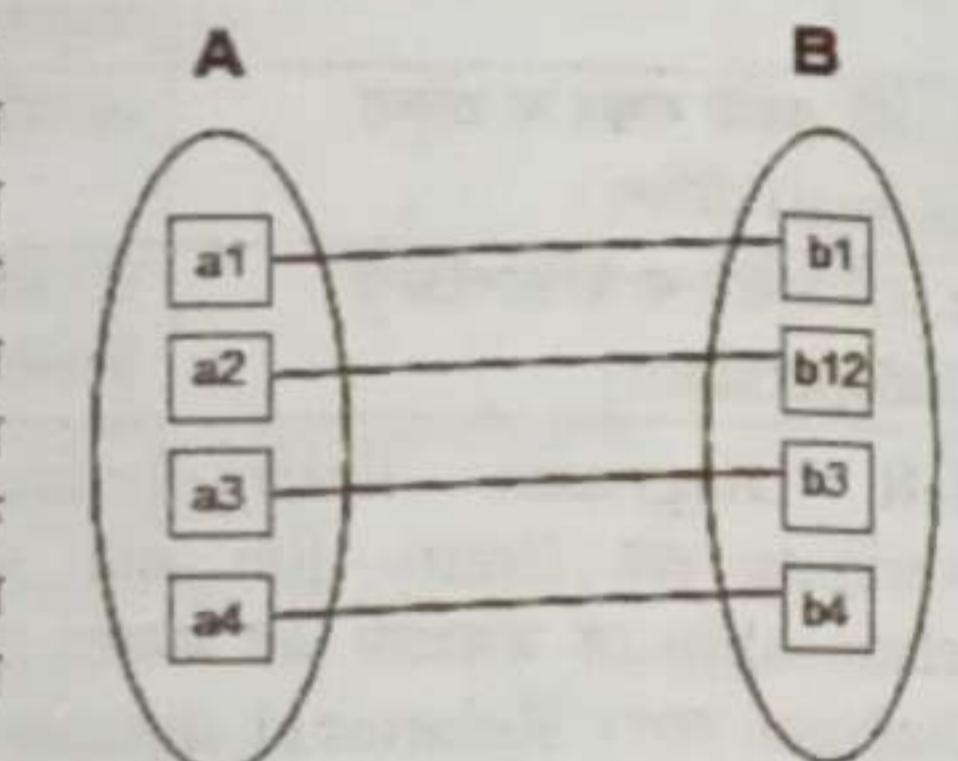
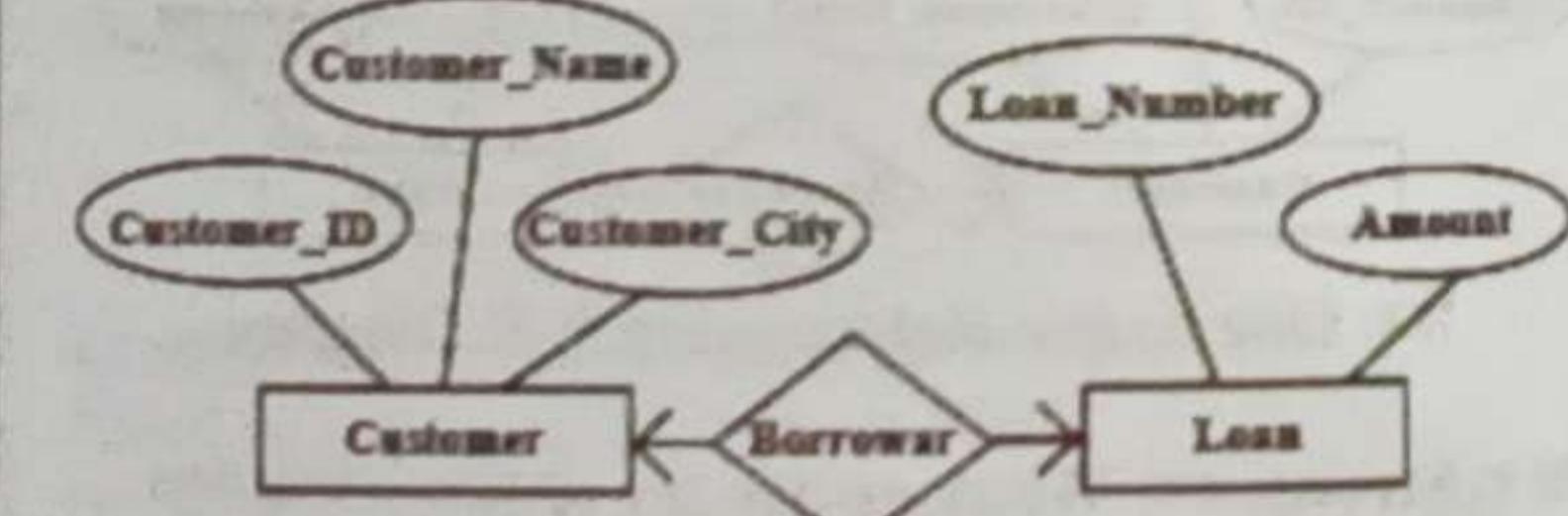


Fig: One to one Relationship

One to one using E-R Diagram : এখনে কাস্টমার এবং Loan দুটি এনটিটি সেট কে Browser রিলেশনশীপ মাধ্যমে উভ এনটিটি সেট এৰ মাধ্যমে রিপ্ৰেজেন্ট কৰা হয়েছে।



চিত্ৰ: One to one Relationship E-R Diagram

b) গোল টু মেনি/ One to Many : [Dept of ICT -2014]

যখন একটি এনটিটি সেট এৰ একটি এনটিটি অপৰ এনটিটি সেট এৰ এক বা একাধিক এনটিটি সাথে রিলেশন ছাপন কৰে তখন তাকে One to Many Relationship বলা হয়।

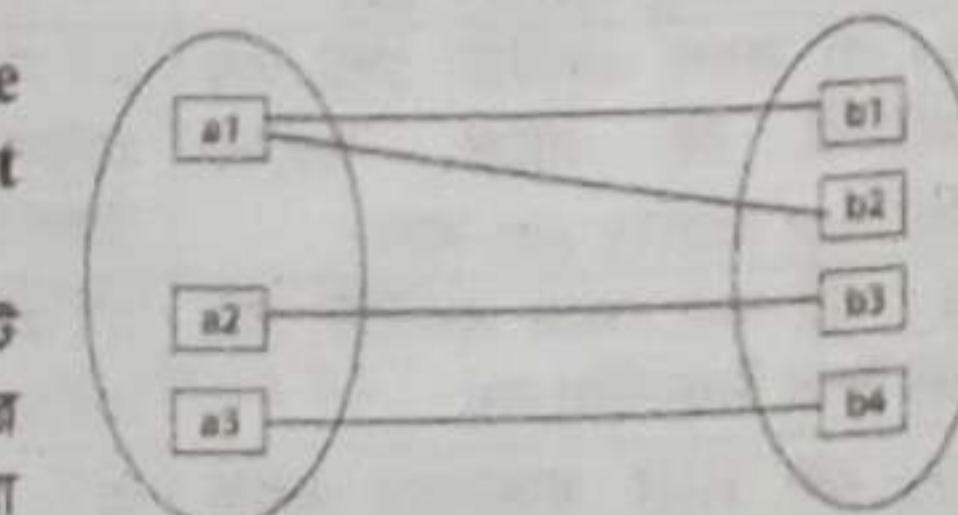
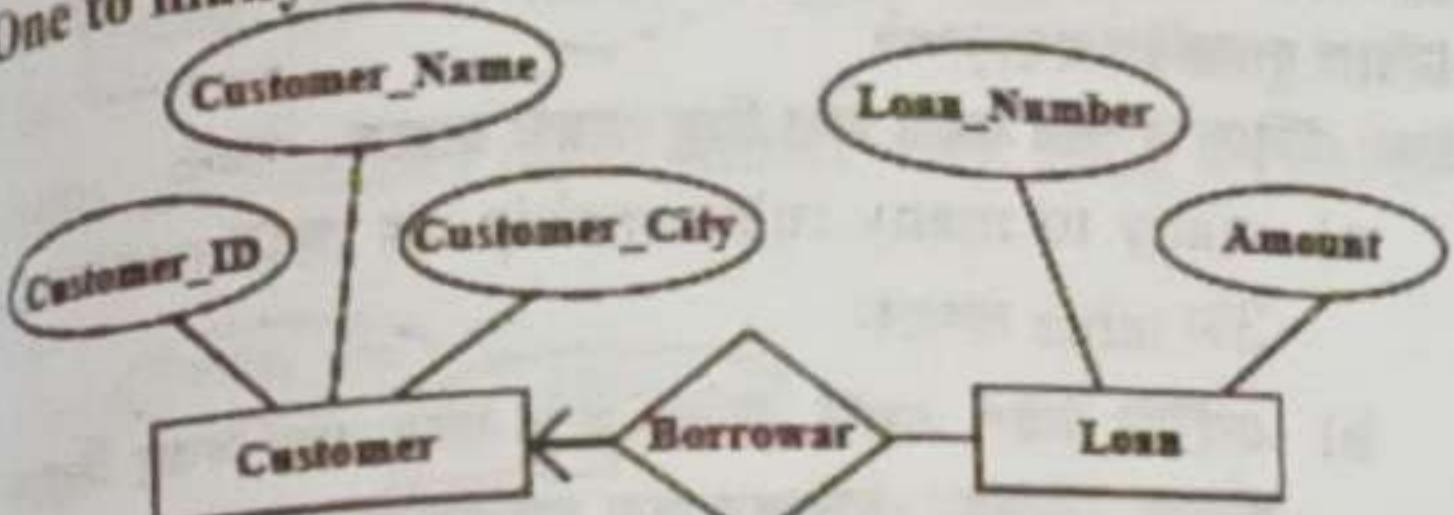


Fig: One to many Relationship

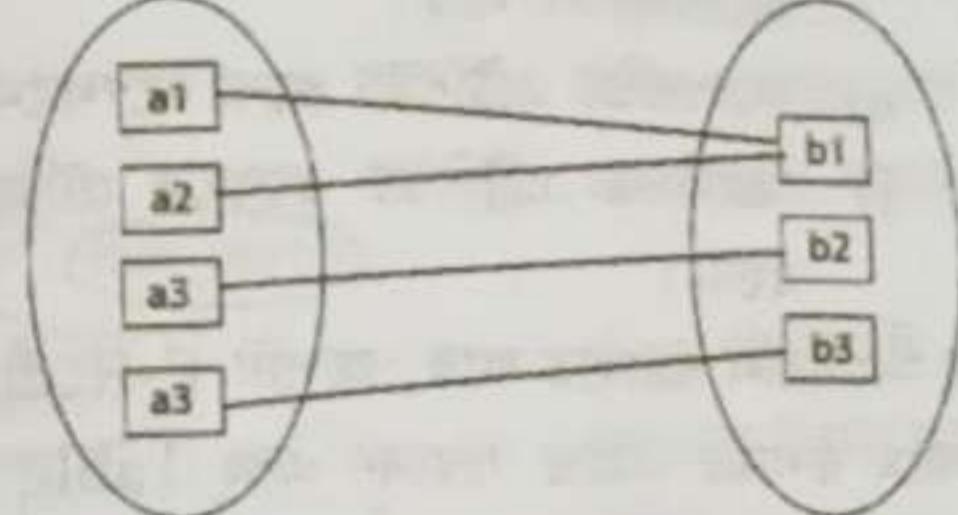
One to many using E-R Diagram :



চিত্ৰ: One to many Relationship E-R Diagram

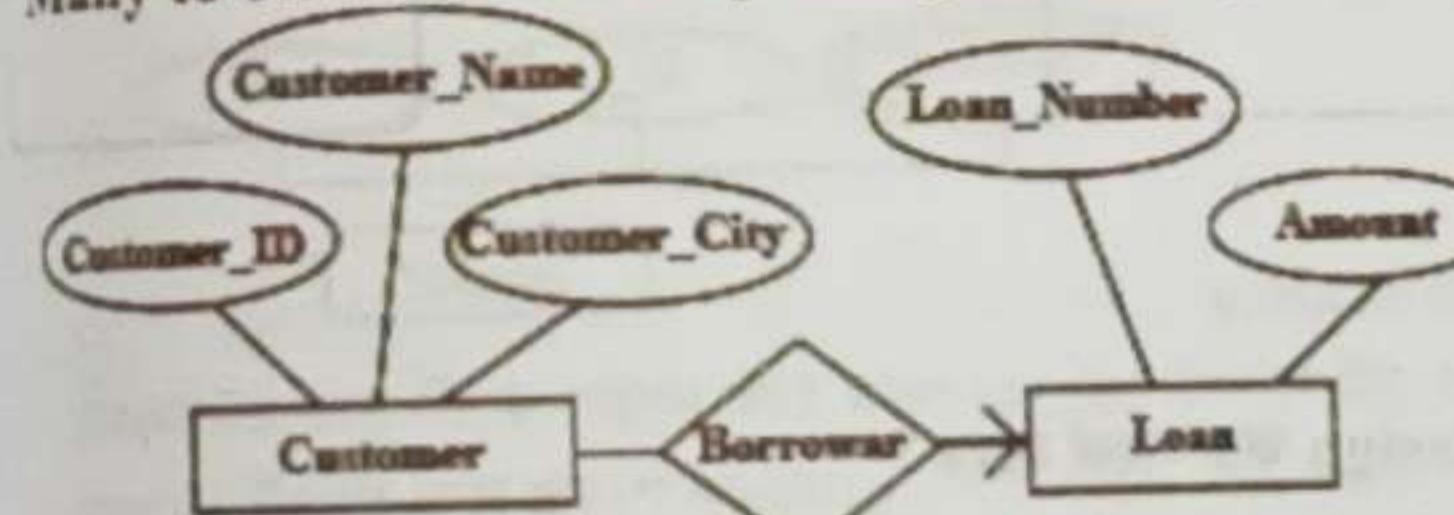
One to one, Many to one, many to many, one to many সহজে E-R Diagram এৰ মাধ্যমে উপহাপনার পদ্ধতি।

c) Many to one:



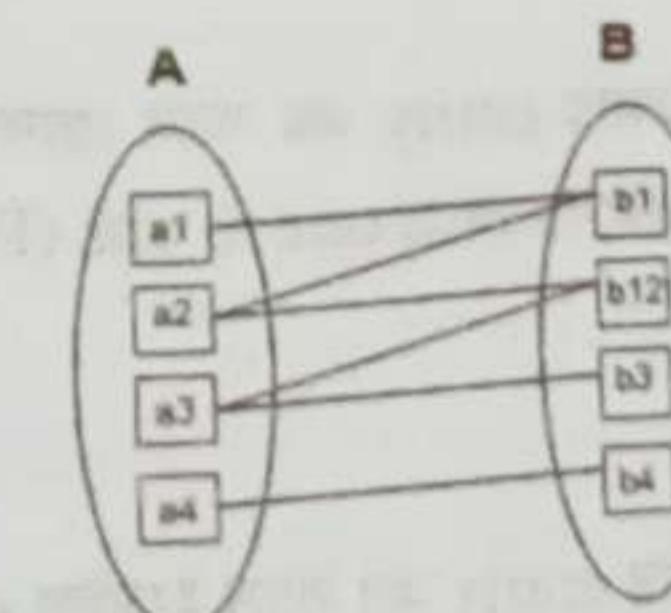
চিত্ৰ: Many to one Relationship

Many to one Relationship using E-R Diagram:



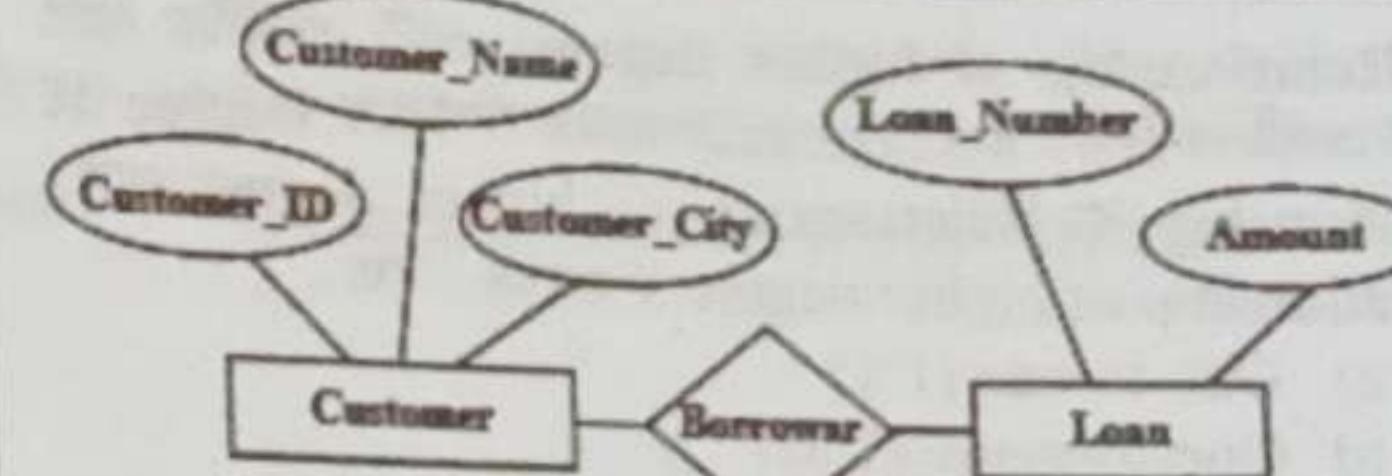
চিত্ৰ: Many to One Relationship using E-R Diagram

d) Many to many : একটি লেট সেট এৰ মধ্যে কোন সংখ্যক এনটিটি অপৰ এনটিটি সেট এৰ মধ্যে নেই সংখ্যক এনটিটি এৰ সাথে সম্পৰ্ক ছাপন মাধ্যমে যে রিলেশন তৈৰি হয় তখন তাকে Many to many বলে।



চিত্ৰ: Many to many Relationship using E-R Diagram

e) Many to many Relation using E-R Diagram:[PSC-2017]



চিত্ৰ: many to many Relationship E-R Diagram

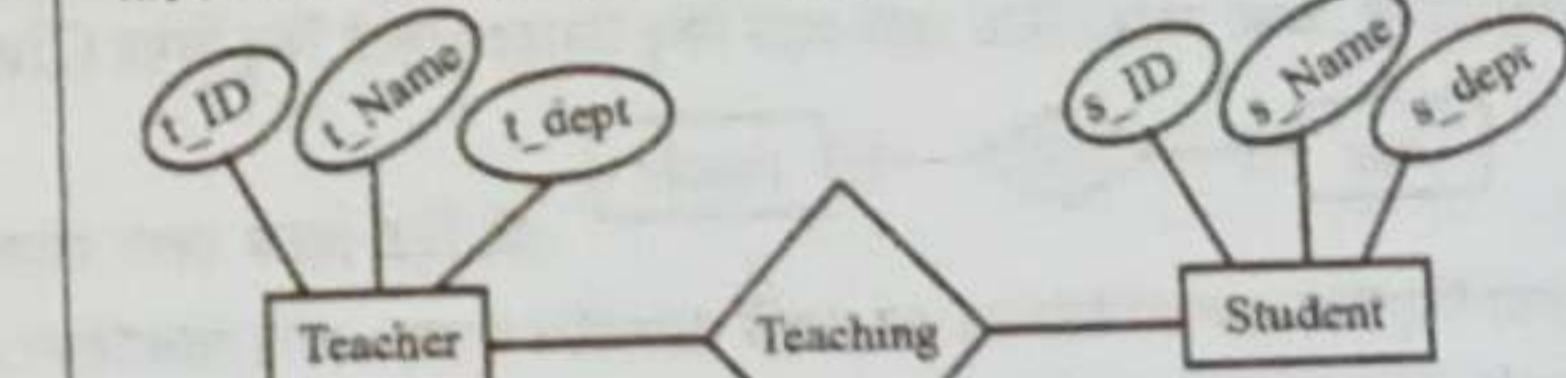
প্ৰশ্ন ১৫. ম্যাপিং কাৰতিনালিটিৰ রিলেশনশিপ তিয়ি আলোচনা কৰ। একটি বাইনাৰি রিলেশনেৰ উদাহৰণ দাও। Discuss the relationship degree of mapping cardinality. Give an example of a binary relation.

উভয়ৰ রিলেশনশিপ : একটি ডেটাবেজেৰ মধ্যে এক বা একাধিক টেবিল থাকে। এই টেবিলগুলোৰ সম্পর্ককে ডেটাবেজ রিলেশনশিপ বলা হয়। ডেটাবেজেৰ রিলেশনশিপ তৈৰি কৰাৰ জন্য যে কয়টি এনটিটি সেট বা টেবিল ব্যবহাৰ কৰা হয় তাৰ সংখ্যাকেই রিলেশনশিপেৰ তিয়ি বলা হয়।

১. ইউনাৰি রিলেশনশিপ
২. বাইনাৰি রিলেশনশিপ
৩. টাৰনাৰি রিলেশনশিপ

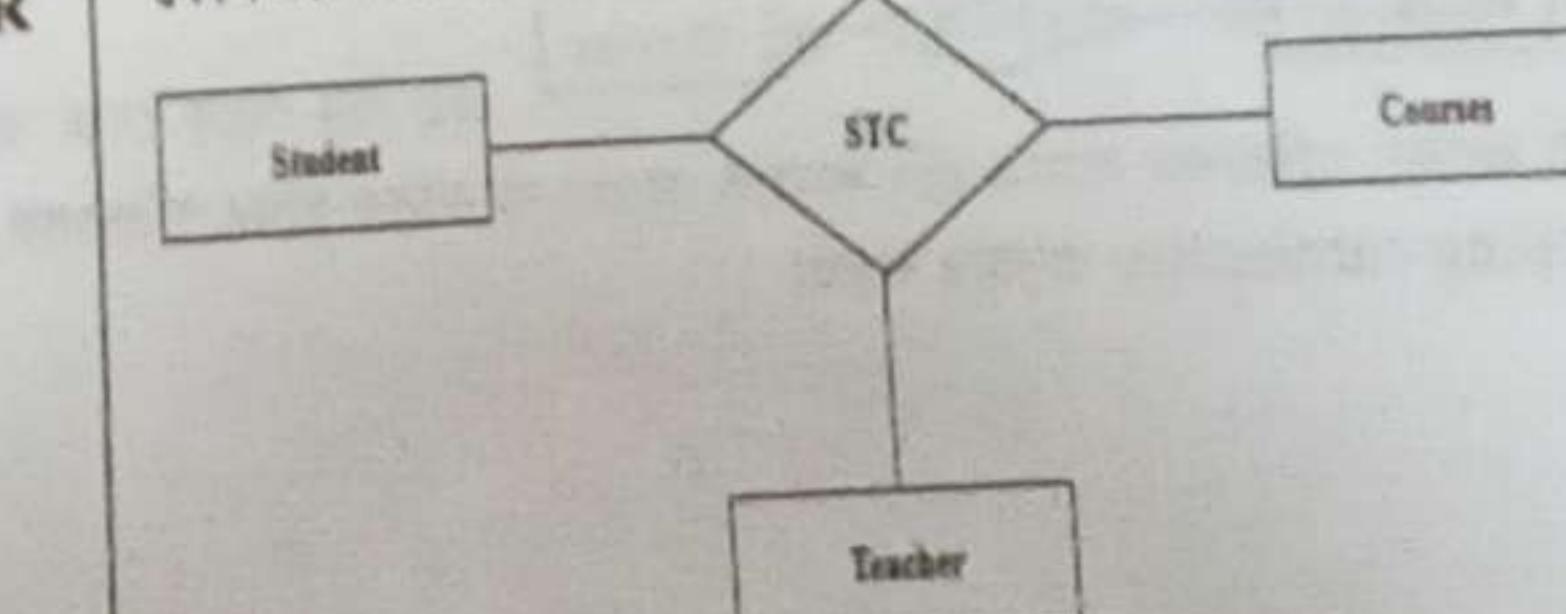
ইউনাৰি রিলেশনশিপ ইউনাৰি রিলেশনশিপে বৰু মাত্ৰ একটি এনটিটি সেট/টেবিল অংশহান কৰে। যেমন- মানুষ একটি এনটিটি। একজন মানুষ অন্য একজন মানুষকে বিয়ে কৰে। কাজেই মানুষ এনটিটি নিজেৰ সাথে নিজেৰ রিলেশনশিপ তৈৰি কৰেছে।

বাইনাৰি রিলেশনশিপ যে রিলেশনশিপ দুটি এনটিটি বা টেবিল অংশহান কৰে তাকে বাইনাৰি রিলেশনশিপ বা ডিমি-২ বলে। উদাহৰণ: শিক্ষক ও ছাত্ৰ দুটি পৃথক এনটিটি বা টেবিল। শিক্ষক ছাত্ৰকে পাঠদান কৰেন। কাজেই এখনে শিক্ষক এনটিটি ছাত্ৰ এনটিটিৰ সাথে বাইনাৰি রিলেশনশিপ বা ডিমি-২ তৈৰী কৰেছে।



চিত্ৰ: বাইনাৰি রিলেশনশিপ বা ডিমি-২

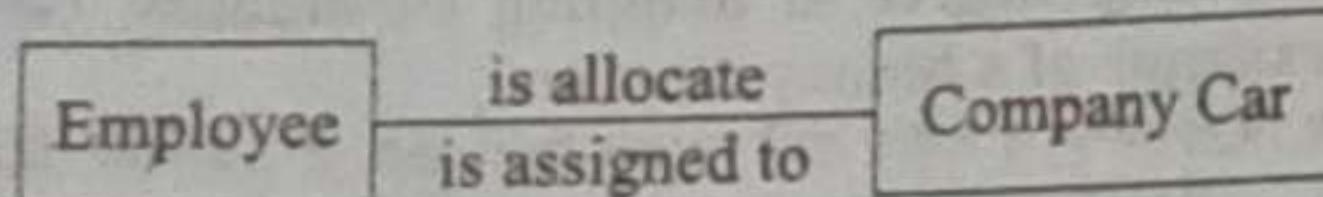
টাৰনাৰি রিলেশনশিপ টাৰনাৰি রিলেশনশিপে তিনটি এনটিটি সেট/টেবিল অংশহান কৰে। যেমন- ছাত্ৰ, শিক্ষক ও কোৰ্স তিনটি পৃথক এনটিটি সেট বা টেবিল। শিক্ষক কোৰ্স নিয়ে থাকেন। কাজেই ছাত্ৰ, শিক্ষক ও কোৰ্স টাৰনাৰি রিলেশনশিপ তৈৰি কৰেছে।



■ Relationship of higher degree: একটি এন্টিটির সাথে অন্য একটি এন্টিটির occurrences সংখ্যার সম্পর্ককে Degree of relationship বা Relationship of higher degree বলে। Relationship of higher degree ৩ ধরনের। যথা -

- a) One-to-one (1:1)
 - b) One-to-many (1:M)
 - c) Many-to-many (M:N)

One-to-one (1:1): একটি এন্টিটির একক বৈশিষ্ট্যের সাথে অন্য একটি এন্টিটির একক বৈশিষ্ট্যের সম্পর্ককে One-to-one রিলেশন বলে। যেমন-



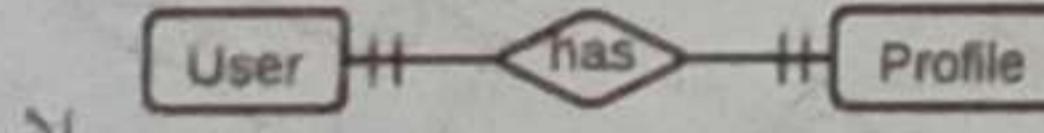
One-to-many (1:M): একটি এন্টিটির একক বৈশিষ্ট্যের সাথে অন্য একটি এন্টিটির অনেকগুলো বৈশিষ্ট্যের সম্পর্ককে One-to-many রিলেশন বলে। যেমন- একটি department এর সাথে Employee সম্পর্ক One-to-many রিলেশন।

Many-to-many (M:M): একটি এন্টিটির অনেকগুলো বৈশিষ্ট্যের সাথে অন্য একটি এন্টিটির অনেকগুলো বৈশিষ্ট্যের সম্পর্ককে Many-to-many রিলেশন বলে। যেমন - Employee এর সাথে project এর সম্পর্ক many-to-many রিলেশন।

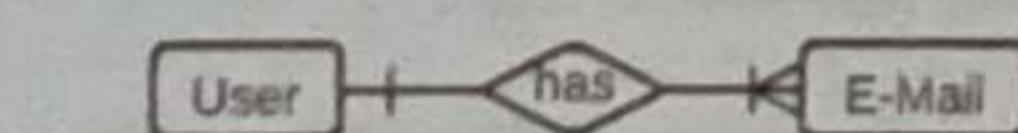
প্রশ্ন ১৬. While converting E-R diagram into Tables, how is a Many-to-many relationship set between entities A and B is converted into database tables? [E-R ভাষ্যকারীকরণে টেবিলে রূপান্তর করার সময়, A এবং B entities

E-R Diagram Design এর ক্ষেত্রে কিছু বিষয় শাখাম রাখলে diagram design করা সহজ হবে। *

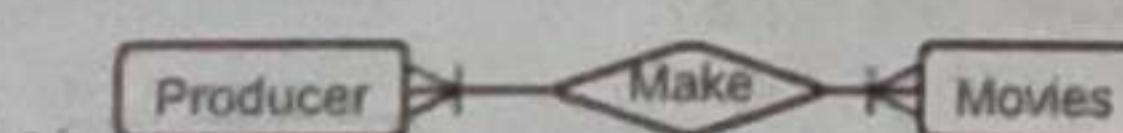
উপরে E-R Diagram এর সকল entity, attribute, relationship, mapping cardinality নিয়ে বলা হয়েছে। ধরে নিলাম সেগুলো আপনাদের জানা আছে। নিচে ছোট ছোট কিছু উদাহরণ দিয়ে কিছু বিষয় Clear করে নিয়ে একটা বড় উদাহরণ দেখবো।



এই চিত্র থেকে দেখা যাচ্ছে ইউজার entity এর সাথে প্রোফাইল entity এর সংযোগ আছে।
একটা সিস্টেমে একজন ইউজার এর একটি প্রোফাইল থাকবে এটাই স্বাভাবিক। এই ধরনের ক্ষেত্রে one to one (Exactly one) ($1:1$) mapping cardinality ব্যবহার করব।



২। এই চিত্র থেকে দেখা যাচ্ছে ইউজার entity এর সাথে ইমেইল entity এর সংযোগ আছে। একজন ইউজার এর একাধিক ইমেইল থাকতে পারে। এই ধরণের ক্ষেত্রে one to many (\rightarrow) mapping cardinality ব্যবহার করব।

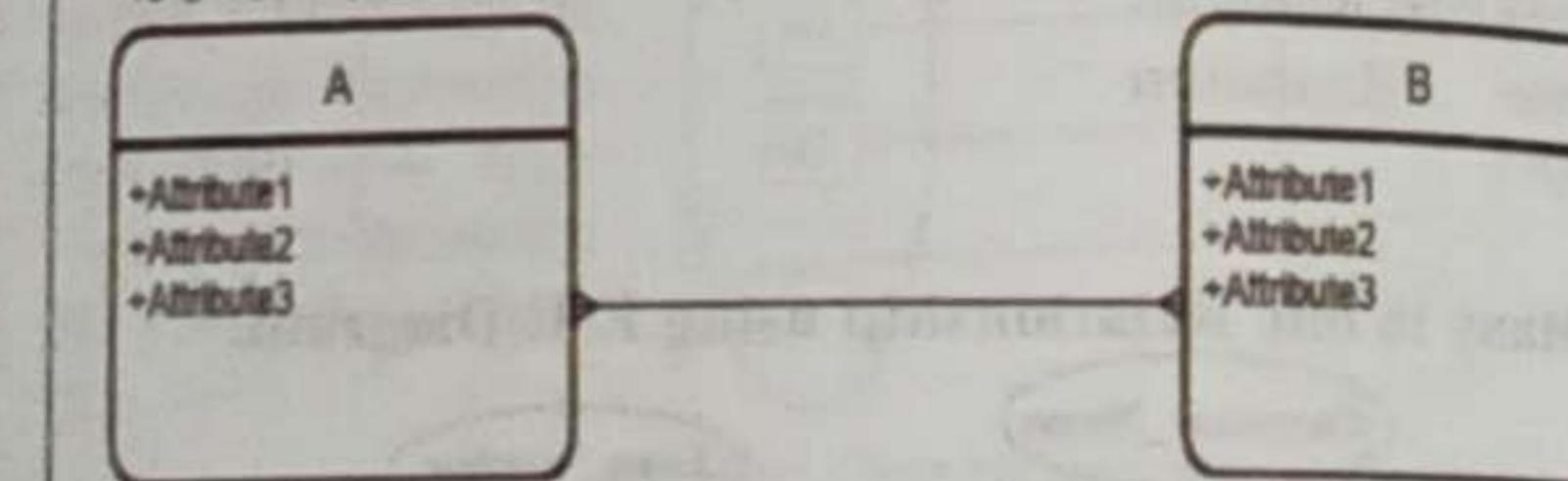


৩।  এই চিত্র থেকে দেখা যাচ্ছে Producer entity এর সাথে Movies entity এর সংযোগ আছে। একজন পরিচালক অনেক ছবি বানাতে পারেন বা অনেক ছবির পরিচালক একজনেই হতে পারে। এই ধরণের ক্ষেত্রে many to many (\bowtie) mapping cardinality ব্যবহার করব।

এর মধ্যে Many-to-many relationship কীভাবে ডাটাকেস টেবিলে সূচাপ্রতি হয়?]***

- a) many to many relationship এর ক্ষেত্রে minimum 3টা table লাগবে।
 - b) এন্টিটি টাইপ একটি টেবিল হয়ে যাবে এবং সকল সিঙ্গেল ভ্যালুড এক্সিবিউট টেবিলের জন্য কলামে পরিনত হবে।
 - c) এন্টিটি টাইপের একটা এক্সিবিউট কী প্রাইমারী কী দ্বারা রিপ্রেজেন্টেড বা প্রকাশিত হবে।
 - d) মাল্টিভ্যালুড এক্সিবিউট তলো একটা আলাদা টেবিল রিপ্রেজেন্টেড হবে।
 - e) কম্পোজিট এক্সিবিউট তলো কম্পোনেন্টস দ্বারা প্রকাশিত হবে।
 - f) ডি঱াইভিউট এক্সিবিউট তলোকে টেবিলের মধ্যে বিবেচনা করা হবে।

এই নিম্নগতি ব্যবহার করে, আপনি E-R diagram কে টেবিল এবং
ক্ষামে রূপান্তর করতে পারবেন এবং Table A এবং Table B এর
মধ্যে যোগাযোগ করতে পারবেন।



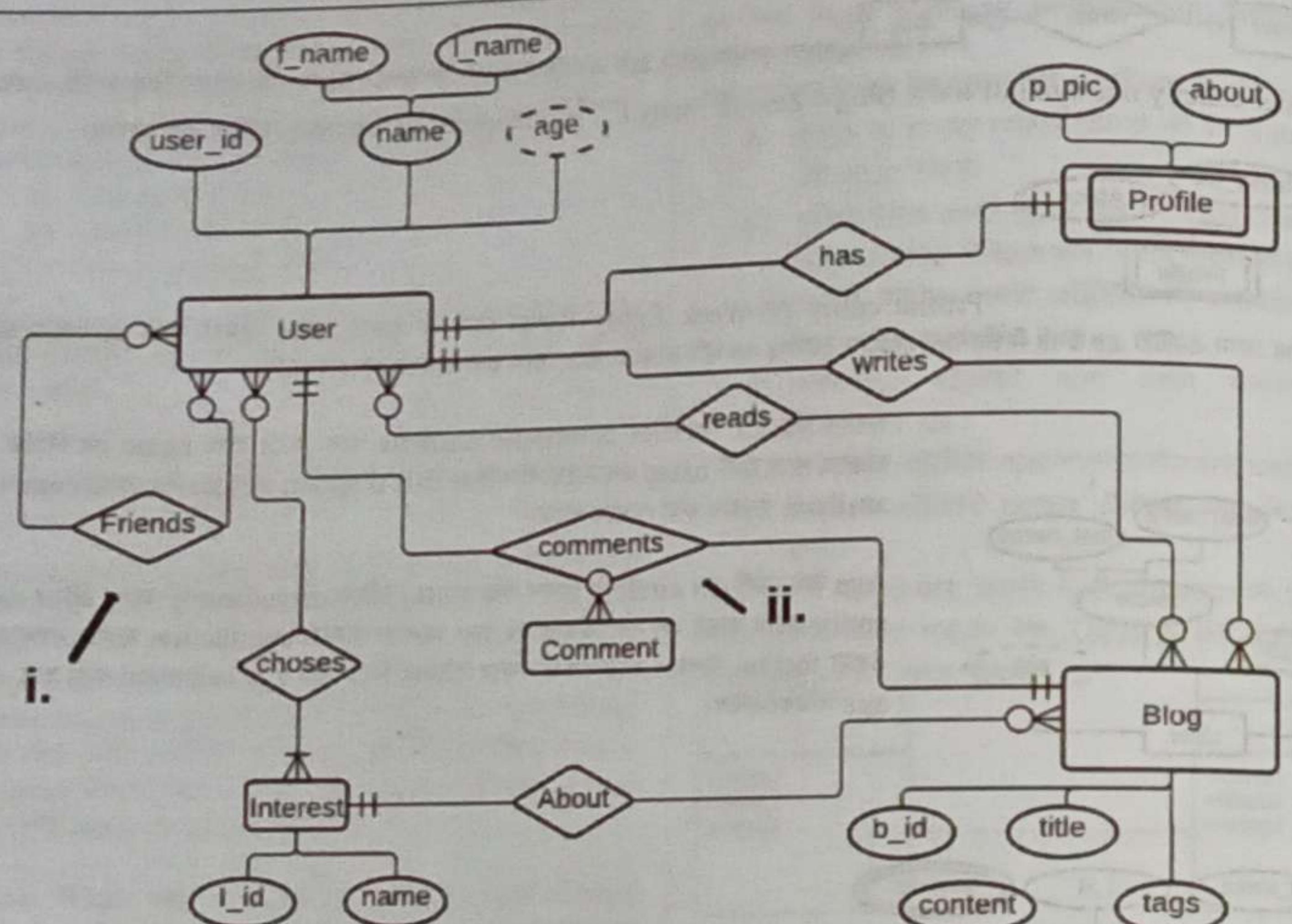
Profile entity কে Week Entity হিসেবে দেখানো হয়েছে কারণ আমরা জানি প্রোফাইল এনটিটির নির্ভরশীল। যেখানে ইউজার এর ছবি থাকতে পারে, তার তথ্য থাকতে পারে ইত্যাদি।

পাশের চিত্রে i. অংশটিকে composite attribute বলে। মানে হলো name এর বিভিন্ন অংশ থাকতে পারে যেটি name এর সাথে সম্পর্কিত। E-R diagram design এর ক্ষেত্রে composite attribute ব্যবহার করা লাগতে পারে।

Age উপাদানটি dot circle এ প্রকাশ করা হয়েছে। এটিকে driven entity বলে। এটিকে driven entity করার কারণ হল সব ইউজার এর জন্য আলাদা আলাদা age function থাকার দরকার নেই। একটা function ব্যবহার করে একজন করে ইউজার নিয়ে তার age calculate করা হবে, যেহেতু age পরিবর্তনশীল।

subject attribute টিকে double উপর্যুক্ত দেয়ার কারণ / কখন নিব? একজন চিচার একাধিক subject উপাদানটির একাধিক ভ্যালু থাকতে পারে। কখনো কোন উপাদানের একাধিক ভ্যালু থাকলে তাকে

ক্ষেত্রটি complete E-R diagram নিচে আলোচনা করা হলঃ***



উপরে একটি অনলাইন ব্লগ এর E-R Diagram আকা হয়েছে। যেখানে একাধিক ইউজার আছে এবং তাদের একটা করে প্রোফাইল আছে, যেহেতু ইউজার এর প্রোফাইল আছে তাই প্রোফাইল হল week এনটিটি (ডাক্ল আইডেক্স) হারা দেখানো হয়েছে। উপরের চিত্রে ইউজার এর friend name এ একটা relation দেখানো হয়েছে (i.) এই ধরনের relationship কে recursive relationship বলে। একটি ব্লগে একজন লগারের বক্তু ও লেখালেখি করতে পারে, তারও একই ধরনের বৈশিষ্ট্য (attribute: first name, last name, user id etc.) থাকতে পারে; কখনো এমন ধরনের রিলেশন দেখা দিলে তখন রিকাসিভ রিলেশন ব্যবহার করা হয়। কমেন্ট ইউজার করে মানে ইউজারের উপর নির্ভরশীল। একজন ইউজার একাধিক কমেন্ট করতে পারে comments relation (ii.) এর নিজস্ব attribute থাকতে পারে; এই ধরনের relationship গুলোকে Ternary Relationship বলে। একটি entity এর সাথে অন্য entity এর mapping cardinality সম্পর্কে অপরে তো আলোচনা করাই হল।

NB: ইউজার/ অর্গানাইজেশনের rules এর উপর ভিত্তি করে mapping cardinality ভিজ হতে পারে।

যেনে রাখা ভাল:

কোন E-R Diagram থেকে Database এ কতগুলো Table হবে তা বের করতে বললে নিচের শর্ত গুলো মনে রাখবেনঃ

- one to many relationship এর ক্ষেত্রে minimum ২টা table লাগবে।
- many to many relationship এর ক্ষেত্রে minimum ৩টা table লাগবে।
- Database এ column গুলো attribute প্রকাশ করে এবং row গুলো entity প্রকাশ করে।

প্রশ্ন ১৭. In a Movie Database have those field, Movie (movie_Id, movie_title , release_date) , Actor (actor_id, actor_first_name, actor_last_name), Director (dir_id, dir_first_name, dir_last_name) and Genre(horror, action, thriller). Now draw an ER diagram of this movie database.

Ans:

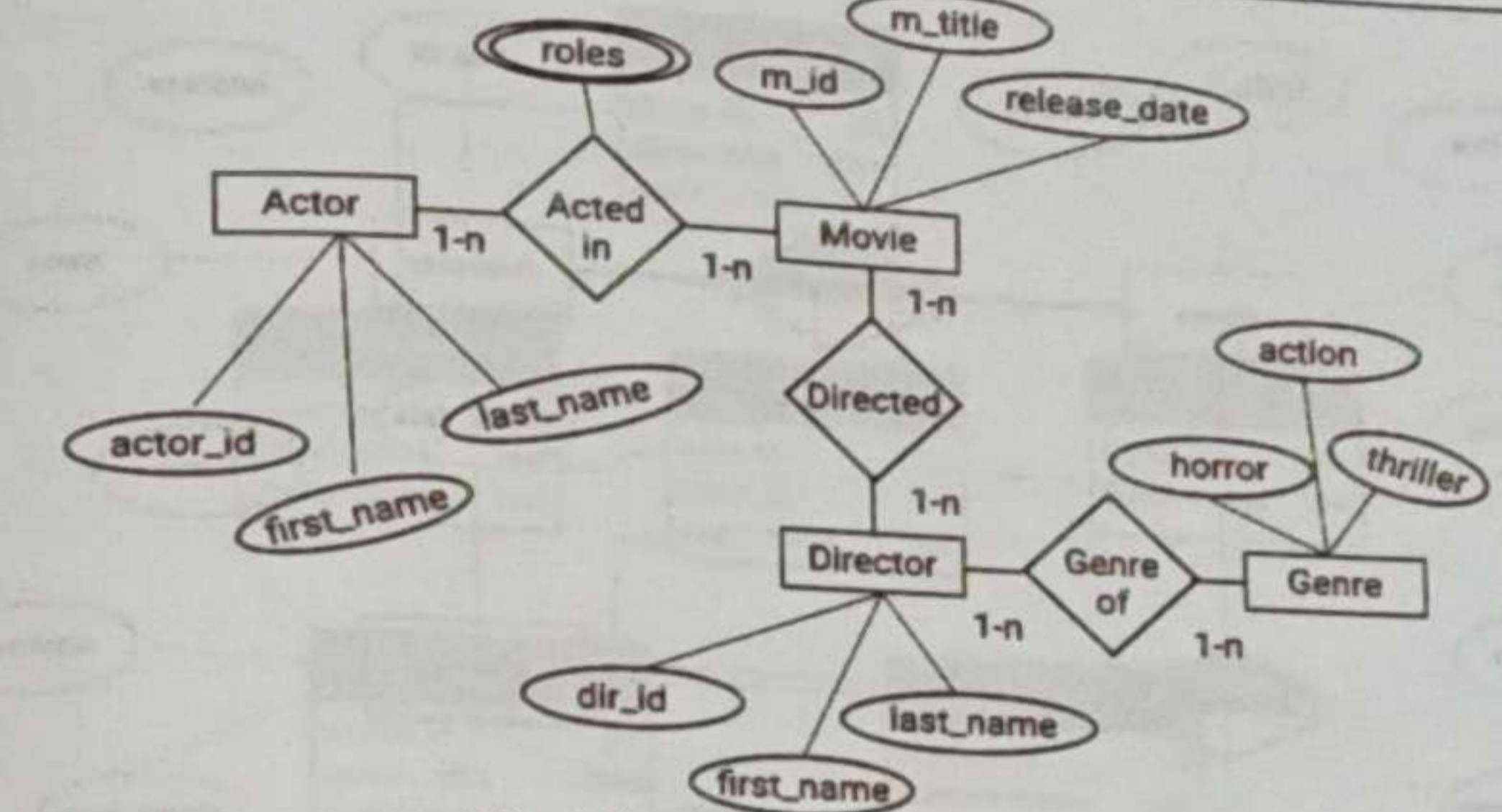


Fig: Movie Database E-R Diagram.

প্রশ্ন ১৮. A book has (id, name, price) can only one author have (id, name, city) and publication have (id, name, city). Book only relation one author and publication are many relations of book. Now draw the E-R diagram.

Ans:

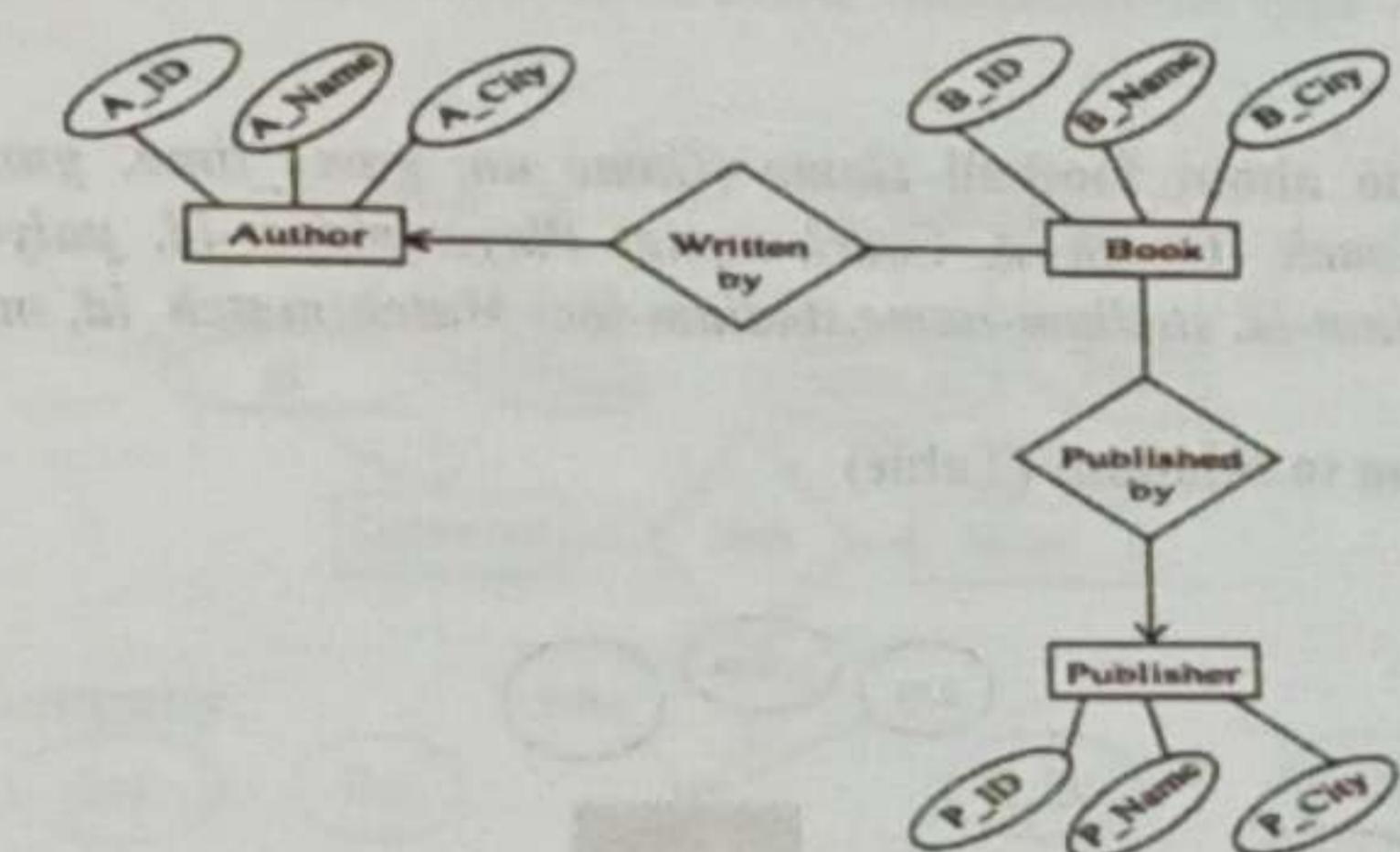


Fig: E-R Diagram

প্রশ্ন ১৯. Draw an E-R diagram of a Library Management System.

Answer:

Database Management System

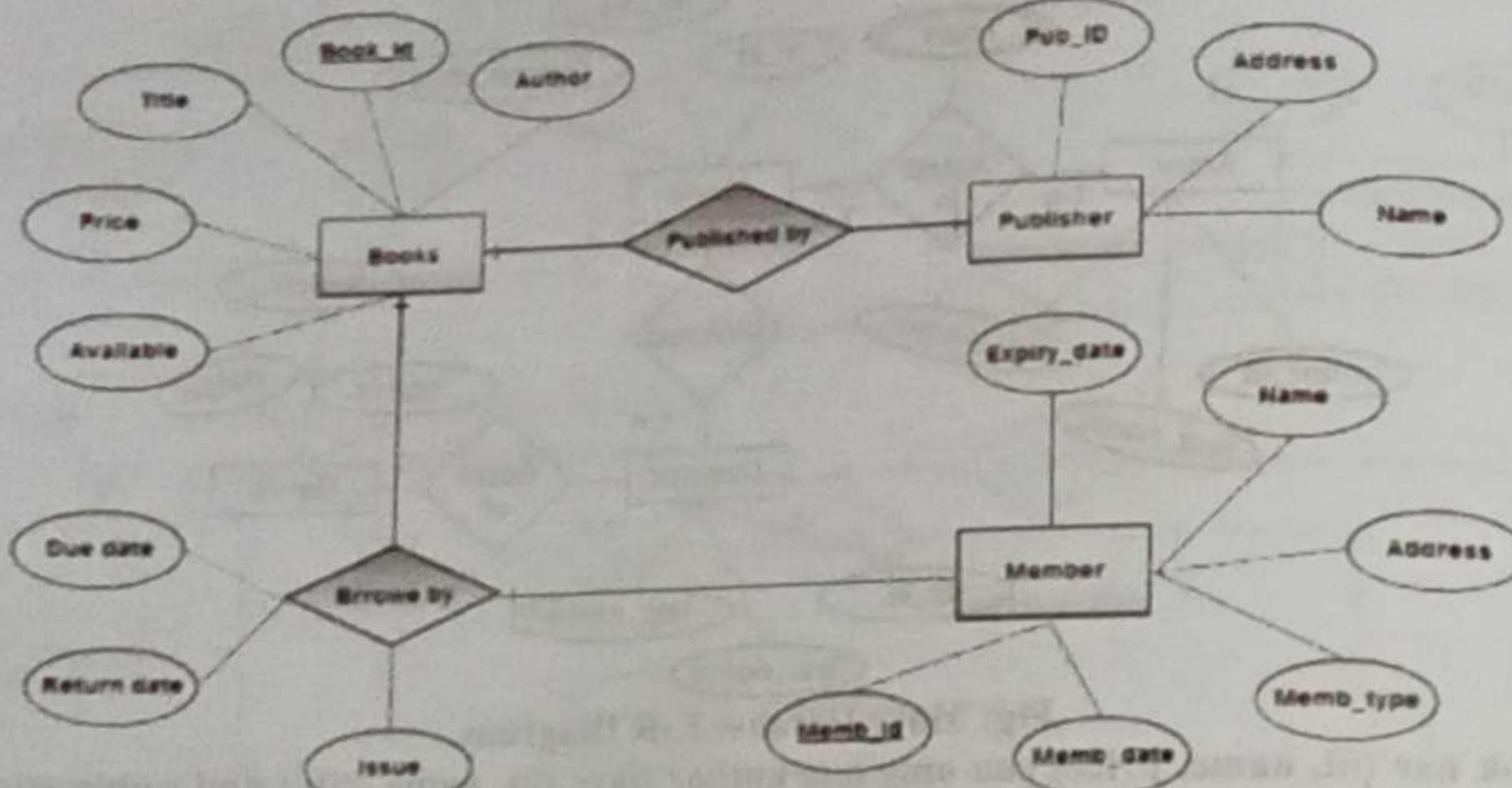


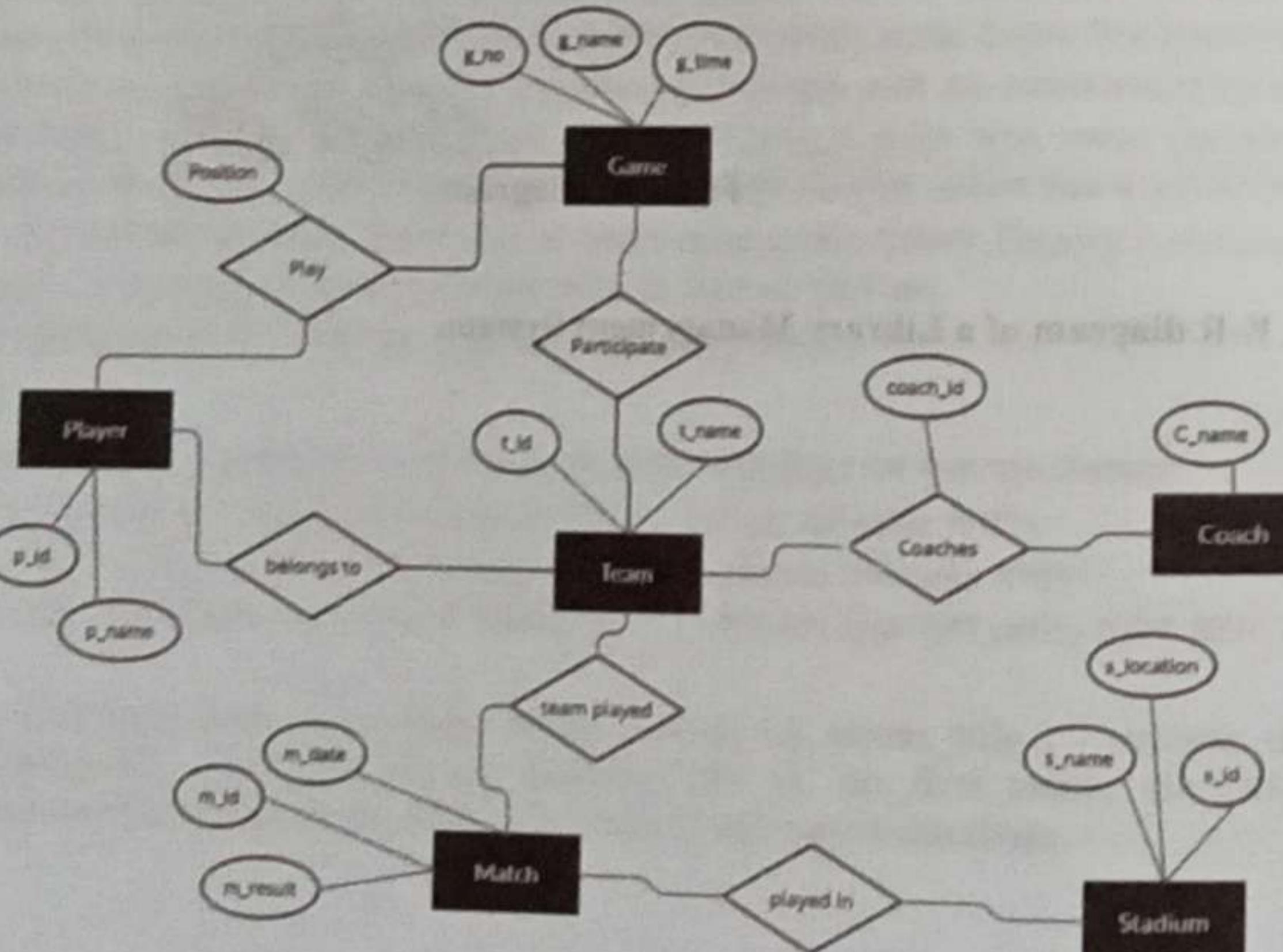
Figure: E-R diagram of a Library Management System

Q20. Given a scenario about football Game (*Game_no*, *game_time*, *game_name*), Team (*team-id*, *coach_id*, *team-name*), Coach (*Coach-id*, *Coach-name*) Player(*player-id*, *palyer-name*, *player-position*), Stadium information(*stadium-id*, *stadium-name*,*stadium-loc*) Match(*match_id*, *match_date*, *match_result*).

i. Draw ER diagram

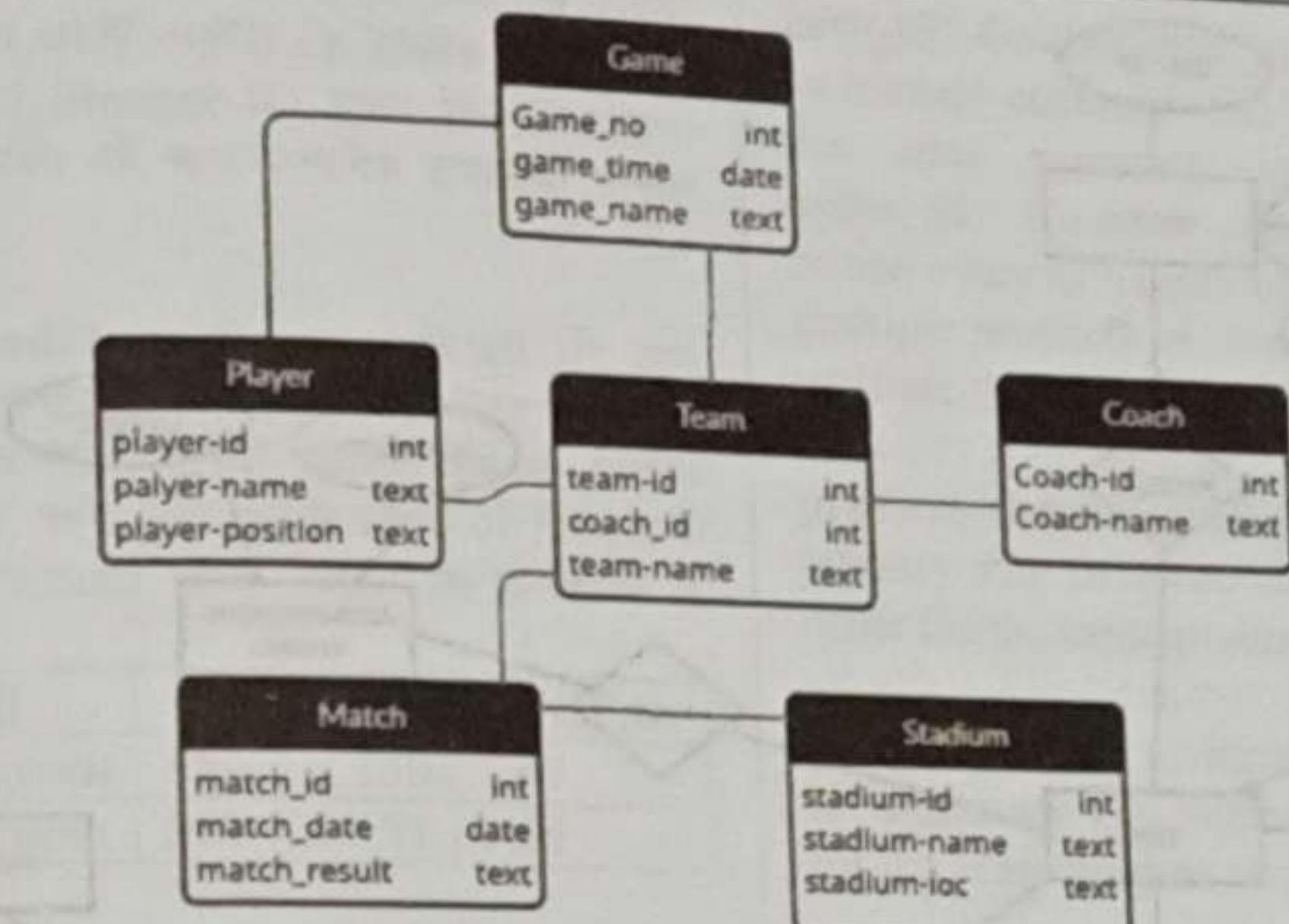
ii. Convert the ER diagram to relations (Table)

Solution: i. ER diagram:



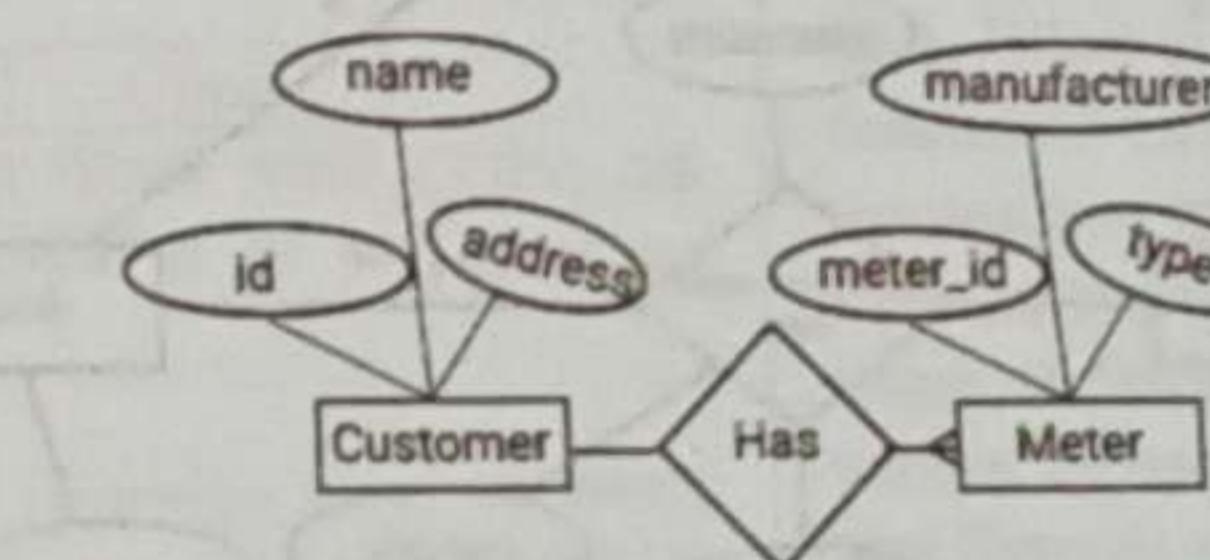
ii. Relational Table:

Database Management System

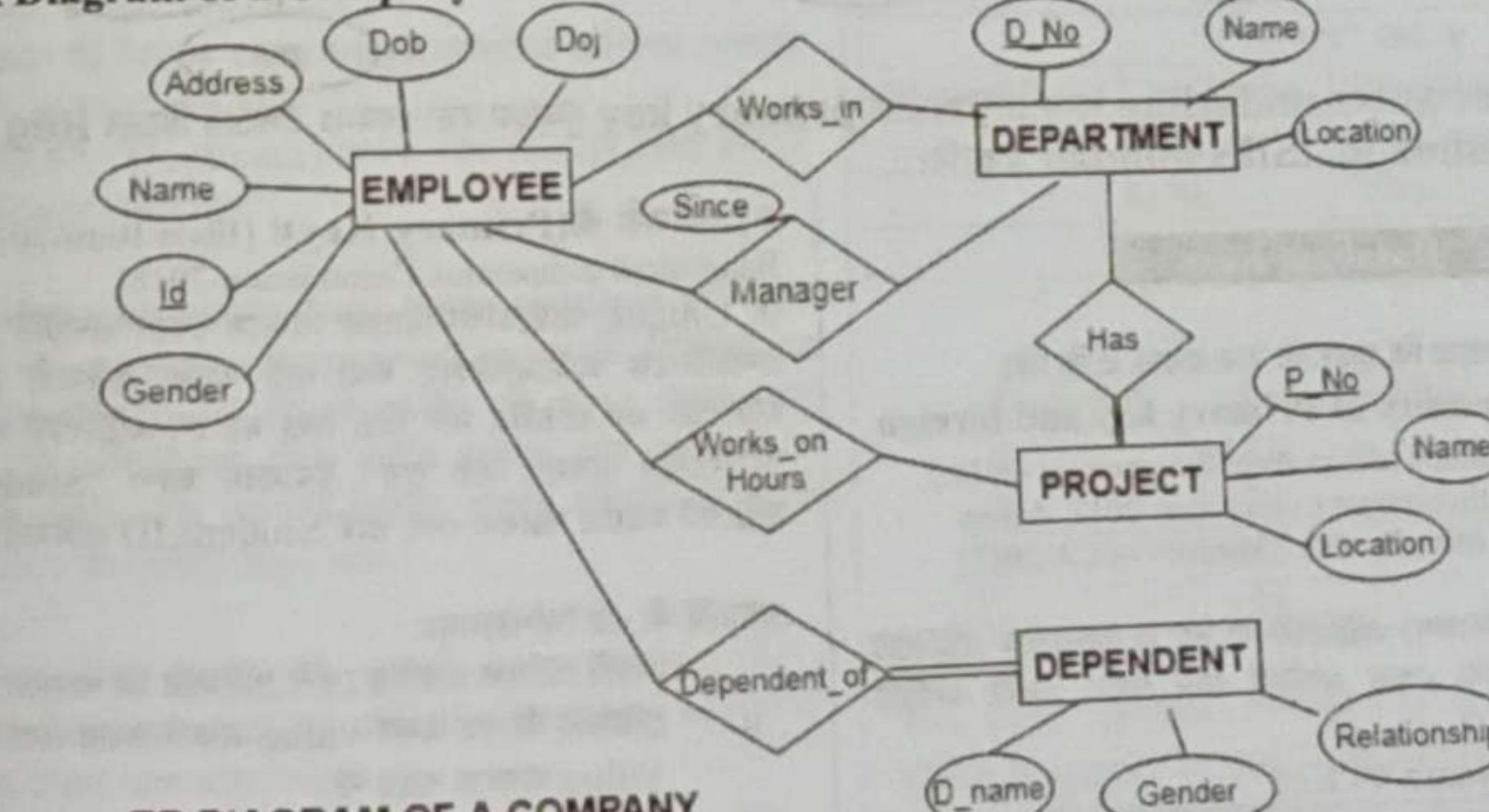


Q21. Titas gas have many customers. A customer has many meters and a meter is owned by only one customer. A customer has id, name and address. A meter has meter-id, type and manufacturer. Draw ERD.

Ans:

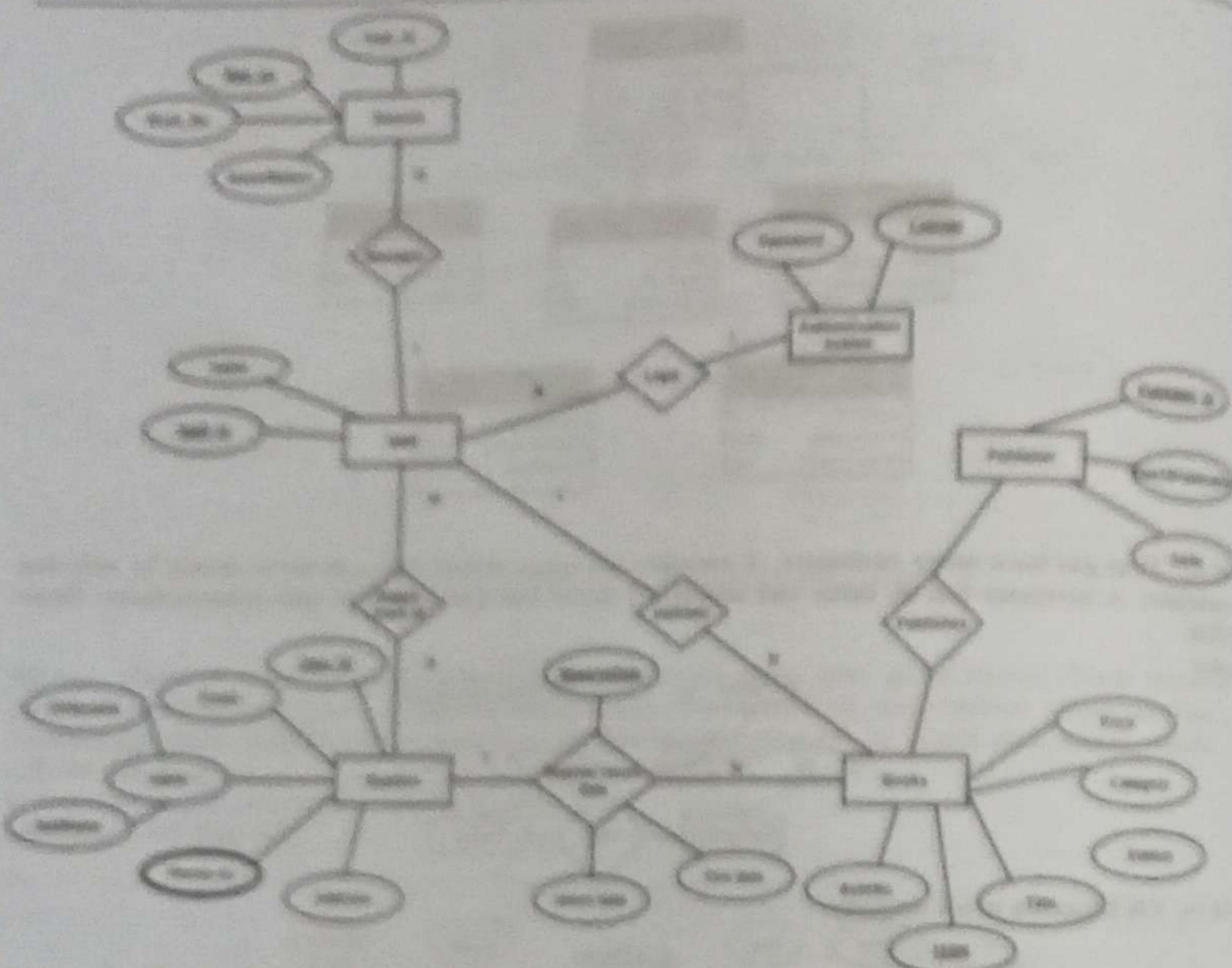


Q22. ER Diagram of a Company.



ER DIAGRAM OF A COMPANY

Q23. ER diagram of Library Management System



Note: In this diagram the underlined are one or primary key pair to cross check Reg. no. digitized publisher at USBN number book.

Q. Database Keys :-

(a) What are the types of keys used in DBMS? Describe the necessity of Primary keys and Foreign keys. (Answer any three sub-questions from Question 201/202/203/204/205/206/207/208/209/210/211/212)

Ans. If we have three different types of primary keys or foreign keys then one will be used with another.

Q. Define what is Primary Key?

Name	ID
Sonam	101
Jasmin	102
Kalpana	103

Q. What is Primary Key? (Jhus Rimu-2018, BCS)

A Unique identifier which can identify each entity in database and can never be repeated in any row. It is also known as atomic key because it can never be divided into two parts. Unique key "Student ID" is composed of two parts i.e. Student ID and Roll No.

Ans. In an example,

- Primary key can't have null values in every row.
- Primary key can't have value mismatched with Null Value present in row.
- Primary key can't have same value in two different rows in same table.

Q. What is Super Key? (Jhus Rimu-2018)

Database Management System

Q. What is an composite key? We will say that a key with which can exist in DBMS are known as composite key. For example if you have a student whose name is "Shivam" and address is "Kalyan". Then student ID will be "Shivam" and "Kalyan" together. So in case all the entries repeat then it is called composite key.

Q. What is Candidate Key? Without it, the data will not have a unique and a permanent value. Different values will be given to different rows. Hence without it, there will be no unique "Name" and "ID" in without it.

ID	Name	CGPA
101	Shivam Kumar	10.0
102	Aditya Kumar	10.0

Q. What is Foreign Key? (Jhus Rimu-2018)

It is often the key that exists in a copy of the R table or part of R table.

Ans. **Student**
Student_ID, Student_name

Q. What is Primary Key? (Jhus Rimu-2018)

If we take a look at R table with CGPA as one of the Student_ID fields or we consider it as a Candidate key.

Ans. In this case,

- we can't have same value (Reference table) placed in more than one row.
- we can't have same value placed in more than one row.
- we can't have same value placed in more than one row.
- if present (Duplicate) then the (Null) can't be inserted.

Q. Explain composite key with example. (Jhus Rimu-2018)

Composite Primary Key: If there are two or more attributes in a row, then one attribute will be unique as well as other because both of them are required to identify the row in this case composite key.

Note: Any key such as super key, primary key, candidate key etc. can be called composite key if it has more than one attributes.

Example: Let's consider a table Sales. This table has four columns (attributes) - cust_id, order_id, product_code, and product_count. None of these columns alone can play a role of key in this table.

Ans. In DBMS over 10 above-mentioned features above a single composite key plays multiple roles. But the same customer can have multiple address under 10 categories under 10 address values in a primary key as it can refer one customer the user of multiple products. Thus same value 10 can be present multiple times.

product order: Customer product code should be a primary key as more than one customer can place order for the same product.

product: Customer product code alone cannot be a primary key because two codes can be placed for the same product code.

Based on this, it is safe to assume that the key should be having more than one attributes. Key in above table (out of product order) has a composite key as it is made up of more than one attributes.

Q. 2. What are purposes of using foreign key in a database? Give suitable examples. (With atleast foreign key 4)(BSCS was down for Google Doodle Day). You will observe that in our DB we need three more R table rows out of two rows that are already present. Doodle gift is one other reason why Google ribbon code was taken out of rows as per Google.

"Player" table			
PlayerID	LastNames	FirstNames	Age
1	Ronaldo	Cristiano	30
2	Lionel	Messi	30
3	Diego	Lamas	28

OrderID	ShoeNumber	PlayerID
1	121	1
2	130	1
3	141	2
4	151	1

Now our "Order" relates PlayerID with "Player" after PlayerID where Row no. 1, 2, 3, 4.

"Player" relates PlayerID with "Player" after PlayerID where Row no. 1, 2, 3, 4.

"Order" after will be 1.

প্রশ্ন ৩. সুপার কী এবং ইউনিক কী এর মধ্যে পার্থক্য লিখুন। Write the difference between Super Key and Unique Key.***

Super Key	Unique Key
সুপার কি হল এক/একাধিক এট্রিবিউট এর একটি সেট যার সাহায্য একটি এন্টিটি সেটের কোন একটি এন্টিটি কে ইউনিক ভাবে চিহ্নিত করা যায়।	UNIQUE constraints একটি column বা একধিক column এ unique value রাখতে শর্ত প্রযোগ করে থাকে। যদি কোন column এ unique constraints থাকে, তার মানে হল এই কলাম কোন duplicate value রাখতে পারবে না।

primary key value অবশ্যই unique হতে হবে।	Foreign key unique না হলেও হবে।	Candidate key value অবশ্যই unique হতে হবে।
শুধুমাত্র একটি primary key থাকতে পারবে একটি টেবিলে।	একাধিক foreign key থাকতে পারবে একটি টেবিলে।	একাধিক candidate key থাকতে পারবে একটি টেবিলে।
Primary key হল clustered index।	Foreign key হল non-clustered index।	Candidate key হল clustered index।

প্রশ্ন ৫. প্রাইমারি কী এবং সুপার কি এর মধ্যে পার্থক্য লিখুন। [Palli sanchay bank-2018, Jibon Bima corporation-18***]

Primary Key	Super Key
প্রাইমারি কী হল একটি single column value যা ডাটাবেজের রেকর্ডকে unique ভাবে identity করার জন্য ব্যবহার করা যায়।	সুপার কি হল এক/একাধিক এট্রিবিউট এর একটি সেট যার সাহায্য একটি এন্টিটি সেটের কোন একটি এন্টিটি কে ইউনিক ভাবে চিহ্নিত করা যায়।
primary key NULL হতে পারে না।	duplicate or unique values থাকতে পারে।
primary key value অবশ্যই unique হতে হবে।	সংজ্ঞা অনুসারে এগুলি unique।

Note: Cluster index is a type of index that sorts the data rows in the table on their key values whereas the Non-clustered index stores the data at one location and indices at another location.

প্রশ্ন ৬. প্রাইমারি কী, ফরেন কী এবং ক্যান্ডিডেট কী এর মধ্যে পার্থক্য লিখুন। Write the difference between primary key, foreign key and candidate key.***

Primary Key	Foreign Key	Candidate Key
প্রাইমারি কী হল একটি single column value যা ডাটাবেজের রেকর্ডকে unique ভাবে identity করার জন্য ব্যবহার করা হয়।	একটি টেবিলের প্রাইমারি কী যদি অন্য টেবিলে ব্যবহৃত হয় তখন এ key কে প্রথম টেবিলের সাপেক্ষে দ্বিতীয় টেবিলের ফরেন কী বলে।	ক্যান্ডিডেট কী হলো একটি column বা একধিক column এ unique value রাখতে শর্ত প্রযোগ করে থাকে। যদি কোন column এ unique constraints থাকে, তার মানে হল এই কলাম কোন duplicate value রাখতে পারবে না।
primary key NULL হতে পারে না।	Foreign key NULL value accept করতে পারে।	candidate key NULL হতে পারে।

॥ Integrity Constraints ॥

প্রশ্ন ১. Integrity Constraint Command কি কাজ করে? [BB-2016]*

উত্তর Integrity Constraint Command এর মাধ্যমে রেফারেন্টিয়াল প্রযোগ করে ডাটাবেস এ ডাটা সংরক্ষণ করা হয়।

প্রশ্ন ২. Data Integrity কত প্রকার ও কি কি? সংক্ষেপে লিখ।

উত্তর Write the two principles of relational database model. [Rooppur Nuclear Power Plant-2018]

উত্তর What are the integrity rules in DBMS?**

উত্তর Data Integrity ও প্রকার। যথাঃ

a) Entity Integrity: একই টেবিলে একই ধরনের দুটি row এরেশের অনুমতি বা অনুমোদন না করাই Entity Integrity।

The entity integrity rule refers to rules the primary key must follow.

i. The primary key value cannot be null.

ii. The primary key value must be unique.

b) Domain Integrity: Domain Integrity এর মাধ্যমে restricted data কে প্রবেশ নিষিদ্ধ করে দেয়া হয়।

c) Referential Integrity: যে নিয়মের কারণে দুই বা ততোধিক টেবিলের মধ্যকার Record সম্মত অবস্থার অবস্থান হতে, তাকে Referential Integrity বলে।

[The referential integrity rule refers to the foreign key. The foreign key may be null and may have the same value but, the foreign key value must match a record in the table it is referring to.]

প্রশ্ন ৩. কোন টেবিলে referential integrity constraint গড় করানোর চেষ্টা করা হলে কী ঘটে? উদাহরণসহ ব্যাখ্যা করুন। What happens when an attempt is made to insert a referential integrity constraint on a table? Explain with examples!

উত্তর You enforce referential integrity by enabling it for a table relationship. Once enforced, Access rejects any operation that would violate referential integrity for that table relationship.

This means that Access will reject both updates that change the target of a reference and deletions that remove the target of a reference. To have Access propagate referential updates and deletions so that all related rows are changed accordingly.

Example of Referential Integrity Employee and Department relationship. If we have dept_id as a foreign key in the Employee table then by using referential integrity constraints we can

avoid creating Employee without department or non-existing department.

In short Referential Integrity makes primary key foreign key relationships viable. Let's first create Employee and Department table with a primary key, foreign key, and referential integrity constraints.

CREATE TABLE Department (dept_id INT NOT NULL,

dept_name VARCHAR(256),

PRIMARY KEY (dept_id)) ENGINE=INNODB;

CREATE TABLE Employee (emp_id INT NOT NULL,

emp_name VARCHAR(256),

FOREIGN KEY (dept_id) REFERENCES Department(dept_id)

ON DELETE CASCADE) ENGINE=INNODB;

Above SQL statements will create both Department and Employee tables. dept_id is now a foreign key in the Employee table.

In this SQL, while creating a foreign key we have specified ON DELETE clause which tells, what needs to do when a record from the parent table is deleted. CASCADE referential action allows to delete or update all matching rows from the child table, after deleting a record in the parent table. This way Referential Integrity preserves data integrity of the relationship.

Let's see How Referential Integrity disallows INSERT and UPDATE for a record in the child table for which there is no matching record in the parent table. To check this Referential Integrity example execute the following MySQL queries :

INSERT INTO Department VALUES (1, "Sales");
INSERT INTO Employee VALUES (101, "Rajeev", 2);

INSERT INTO Employee VALUES (101, "Rajeev", 2);
ERROR 1452 (23000): Cannot ADD OR UPDATE a child row: a FOREIGN KEY constraint

`fails ('test', 'employee', CONSTRAINT 'employee_ibfk_1'
FOREIGN KEY ('dept_id') REFERENCES 'department'
('dept_id') ON DELETE CASCADE)`

When we inserted first record in Department table it ran fine but when we insert a record in Employee table with dept_id = 2 which is not present in Department i.e. parent table, failed to Referential integrity or foreign key constraint check.

If you modify your query and correct dept_id to 1, query will run fine, as shown below

```
INSERT INTO Employee VALUES (101, "Rajeev",  
, 1);  
Query OK, 1 row affected (0.05 sec)
```

Now let's delete our only record from Department table and see if matching records on child table is automatically deleted or not.

```
mysql> DELETE FROM Department;  
Query OK, 1 row affected (0.05 sec)
```

```
mysql> SELECT * FROM Employee;  
Empty SET (0.00 sec)
```

You see there is no record in the Employee table because of ON DELETE CASCADE, matching records in the child table are deleted. Similarly, you can use ON UPDATE CASCADE to automatically propagate UPDATE from the parent table to child tables.

প্রশ্ন 8. DBMS কীভাবে 'Data Integrity' সমস্যা দূরীভূত করে?**

উভয় কম্পিউটার সিস্টেম এ জ্ঞাকৃত সর্বমোট স্ট্রাকচারড ডাটার সমষ্টিকে ডাটাবেস বলে। আব স্টোর কৃত Data কে একসেস করার জন্য ব্যবহৃত এক সেট প্রোগ্রাম এর সমষ্টিকে ডাটাবেস ম্যানেজমেন্ট সিস্টেম (DBMS) বলে।

ইন্টিগ্রিটি অস্বীকৃত ডাটাবেসে জ্ঞাকৃত ডাটাকে কিছু কিছু Consistency Constraints পালন করতে বাধ্য করা হয়। যেমন:

- ফাইল ফরমেট ভিন্ন না হয়।
- numeric column এ যাতে alphabetic data না আসে etc.

Consistency —এর বাংলা অর্থ সামঞ্জস্য বা সঙ্গতি অথবা হিল। রিলেশনাল ডাটাবেজ ম্যানেজমেন্ট সিস্টেমে কলিস্টেন্সি দ্বারা আমরা বুঝ যে প্রতিটি ডাটাবেজ ট্রানজেকশনকে ডাটাবেজে নির্ধারিত নিয়মের (Database Constraint) সাথে সামঞ্জস্য রেখে ডাটা পরিবর্তন বা নতুন ডাটা যোগ করতে হবে। আমরা নানা বিধি উপায়ে ডাটাবেজ ট্রানজেকশনের উপরে বাধ্যবাধকতা বা নিয়ম (Database Constraint) নির্ধারণ করতে পারি। যেমন, প্রাইমারি কি (primary

key), ফরেন কি (foreign key), ট্রিগার (trigger), ইত্যাদি যারা আমরা ট্রানজেকশনের উপরে বাধ্যবাধকতা বা নিয়ম আরোপ করতে পারি। মনে করি আমাদের একটি Student টেবিল আছে এবং studentId হল এই টেবিলের প্রাইমারি কি। আমরা যখন নতুন একজন স্টুডেন্টের ডাটা যোগ (data insert) করতে যাব তখন ডাটাবেজ পরীক্ষা করে দেখবে যে প্রাইমারি কি নিয়মটি মানা হচ্ছে কিনা। এভাবেই ডাটাবেজে নির্ধারিত নিয়মগুলো (Database Constraint) প্রতিটি ট্রানজেকশনের সময় পরীক্ষা করে দেখে যেন ট্রানজেকশনটি নিয়মের ব্যূহ না ঘটিয়ে সম্পূর্ণ হয়।

॥ Functional Dependencies ॥

প্রশ্ন 1. Functional Dependency কি? সংক্ষেপে লিখুন।

উভয় ফাংশনাল ডিপেন্ডেন্সি একটি ডাটাবেস ম্যানেজমেন্ট সিস্টেম এ একটি আট্রিবিউটের সাথে অন্য একটি আট্রিবিউটের সম্পর্ক নির্ধারণ করে। এটি সাধারণত প্রাইমারী কি এর সাথে non-key attributes এর রিলেশনকে represent করে। ইহা একটি ডাটাবেস সিস্টেমের উন্নত মান বজায় রাখতে ফাংশনাল ডিপেন্ডেন্সি ক্ষেত্ৰপূর্ণ ভূমিকা পালন করে। ফাংশনাল ডিপেন্ডেন্সি \rightarrow অ্যারো দ্বারা তিনিং করা হয়। যদি Y ফাংশনালী X এর উপর নির্ভর করে তখন এদের লিখা হয় X \rightarrow Y, যেখানে X একটি primary কি এবং non-key attributes সেট Y।

■ Rules of Functional Dependencies:

Reflexive rule: যদি X set of attributes হয় এবং Y X এর subset হয়, তাহলে X Y এর ভ্যালু ধারণ করে।

Augmentation rule: যখন Y ফাংশনালী X এর উপর নির্ভর/ধারণ করে ($X \rightarrow Y$) এবং C যদি আলাদা attribute set হয় তাহলে $XC \rightarrowYC$ এটি সম্ভব, কারণ কিছু অগ্রিমত্ত (C) attribute শুধু যোগ করা হয়েছে basic dependencies পরিবর্তন না করে।

Transitivity rule: কিছুটা বীজগণিতের transitive rules এর মত, যদি $X \rightarrow Y$ এবং $Y \rightarrow Z$ হয়, তাহলে $X \rightarrow Z$ লেখা যাব।

প্রশ্ন 2. Functional Dependencies এর প্রকারভেদ উন্নয়ন সহ সংক্ষেপে লিখুন।

উভয়

- ✓ Multivalued dependency
- ✓ Trivial functional dependency
- ✓ Non-trivial functional dependency
- ✓ Transitive dependency

Multivalued dependency: একটি রিলেশনের যখন এক বা একাধিক Independant value থাকে তখন ঐ রিলেশনে Multivalued dependency সৃষ্টি হয়। মাল্টিভালিউড dependency দুটি attribute রিলেশনের বড় সীমাবদ্ধতা।

Car Model	Manufacture year	Color
H001	2018	Metallic
H001	2018	Green
H005	2019	Metallic
H005	2019	Blue
H010	2017	Metallic
H033	2020	Gray

উভয় যদি কোন এন্টিটি সেট এর একটি এন্টিটির অঙ্গ অথবা এন্টিটি সেট এর অপর এন্টিটি এর উপর নির্ভর করে তখন তাকে একজিস্টেল বলে। উদাহরণ: Loan (x) এবং Payment (y) যেখানে y নির্ভর করে X এর উপর। সূতৰং X কে বলা হয় Subordinate entity এবং Y কে বলা হয় ভোমিনেট এন্টিটি।

প্রশ্ন 8. একটি স্কুলের জন্য তৈরি ডাটাবেজে কি কি তথ্য কিভাবে রাখা যেতে পারে বাধ্যবাধক আলোচনা করুন? [Discuss at a table with an explanation how information can be stored in a database created for a school.]

উভয় নিম্নে একটি স্কুলের ডাটাবেস দেওয়া হয়েছে, এখানে কিছু টেবিল থাকবে তা হলো:

Student : এই টেবিলে স্টুডেন্ট সংক্রান্ত যেমন আইডি, নাম, ডেট অব বার্ষ, মোবাইল, ইমেইল ইত্যাদি থাকবে। স্টুডেন্ট টেবিলের সাথে প্যারেন্ট টেবিলের সাথে একটা রিলেশন আছে এবং Classroom Student, Exam_result, Attendance টেবিলের সাথে রিলেশন আছে।

Parent: এই টেবিলে প্যারেন্ট সংক্রান্ত যেমন আইডি, ফাস্ট নাম, লাস্ট নাম, ডেট অব বার্ষ, মোবাইল, ইমেইল, পাসওয়ার্ড ইত্যাদি থাকবে।

Course: এই টেবিলে Course সংক্রান্ত যেমন আইডি, নাম, ডিপ্লোম, গ্রেড ইত্যাদি থাকবে। Course টেবিলের সাথে Exam_result টেবিলের একটা সাথে রিলেশন আছে।

Grade : এই টেবিলে Grade সংক্রান্ত যেমন আইডি, নাম, ইত্যাদি থাকবে। Course, Classroom টেবিলের সাথে Grade, Classroom_Student, Exam_result টেবিলের একটা সাথে রিলেশন আছে।

Classroom: এই টেবিলে Classroom সংক্রান্ত যেমন আইডি, ইয়ার, গ্রেড, সেকশন, স্ট্যাটাস ইত্যাদি থাকবে। Classroom টেবিলের সাথে Grade, Classroom_Student, Exam_result টেবিলের একটা সাথে রিলেশন আছে।

Classroom_Student: এই টেবিলে Classroom আইডি, স্টুডেন্ট আইডি ইত্যাদি থাকবে।

Attendance: এই টেবিলে Attendance সংক্রান্ত যেমন ডেট, টাইম, স্টুডেন্ট আইডি, স্ট্যাটাস ইত্যাদি থাকবে।

Exam: এই টেবিলে Exam সংক্রান্ত যেমন আইডি, নাম, টাইম, টাইপ, স্ট্যাটাস ইত্যাদি থাকবে।

Teacher: এই টেবিলে Teacher সংক্রান্ত যেমন আইডি, নাম, ডেট অব বার্ষ, মোবাইল, ইমেইল ইত্যাদি থাকবে। Classroom টেবিলের সাথে Teacher টেবিলের সাথে একটা রিলেশন আছে।

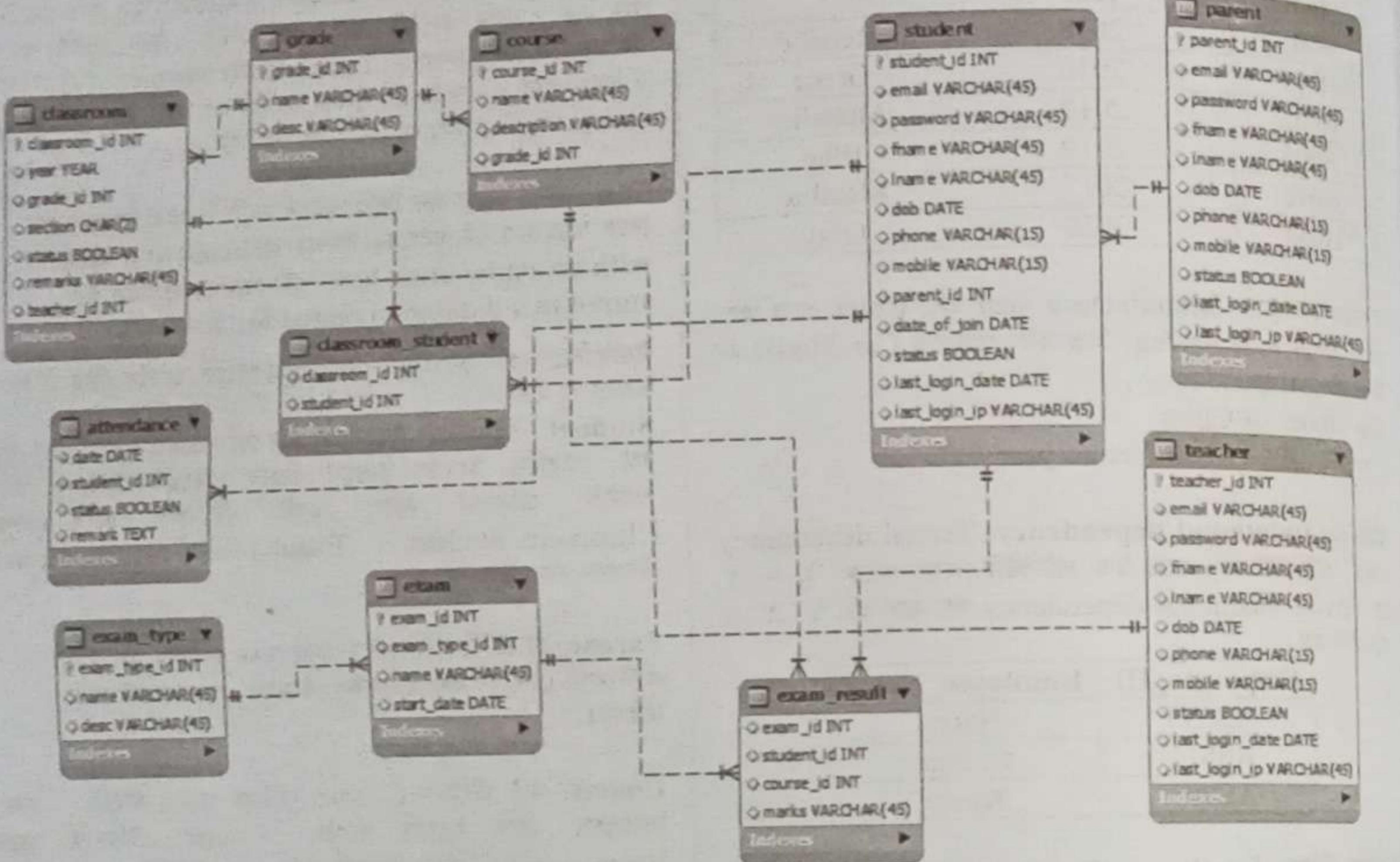
Exam_result: এই টেবিলে Exam_result সংক্রান্ত যেমন স্টুডেন্ট আইডি, Course: নাম, স্ট্যাটাস ইত্যাদি থাকবে।

Company	CEO	Age
Microsoft	Satya Nadella	51
Google	Sundar Pichai	46
Apple	Tim Cook	57

{Company} \rightarrow {CEO} যদি Company নাম জানা যাব তবে CEO নামও জানা যাবে। কিন্তু CEO, Company এর সাব-সেট না। সুতরাং এটি একটি Non Trivial functional dependency।

Transitive dependency: যদি $A \rightarrow B$ এবং $B \rightarrow C$ হয়, অব $A \rightarrow C$ । সুতরাং এটি একটি Transitive dependency।

প্রশ্ন 9. Existence Dependency বলতে কি বুঝ। ভোমিনেট ও Sub-ordinate entity কি?



File Organization

প্রশ্ন ১. File কি?

উভয় পরম্পরাগত কটজলো রেকর্ডের সমষ্টিকে ফাইল বলে যা, বিভিন্ন সেকেন্ডারী স্টোরেজ দ্বারা ম্যাগনেটিক ডিস্ক, চৌম্বকীয় কিন্তু এবং অপটিকাল ডিস্ককে এ রেকর্ড করা যায়।

প্রশ্ন ২. ফাইল অগ্রন্থাইজেশন কি? বিভিন্ন প্রকার ফাইল অগ্রন্থাইজেশন নাম লিখুন উদয়ারণ সহ বর্ণনা করুন। What is file organization? Write different types of file organization names and give examples.

উভয় একটি ফাইল এ যেভাবে উভয় রেকর্ড সবুজ সুসংগঠিত থাকে তাকে ফাইল অগ্রন্থাইজেশন বলে।

বিভিন্ন প্রকার ফাইল অগ্রন্থাইজেশন নাম:

File Organization

- Sequential FO
- Heap FO
- Hash FO
- ISAM
- B+ Tree FO
- Cluster FO

■ **Sequential file organization :** যদি কোন ফাইল তার অনুরূপ, তবে একটি দুটিরও বেশি children থাকতে পারে। এই

করে সাজানো থাকে তবে সেই ফাইল অগ্রন্থাইজেশন কে Sequential file organization বলে।

■ **Heap File Organization:** এটি file organization সহজ উপায়। যখন নতুন কোন record insert করা হয় তখন records গুলো ফাইলের শেষে সংযুক্ত হয়। sequentially method এর মতই কিন্তু এখানে data block কে sequentially selecte করা হয়।

■ **Hash file organization :** Hashing হলো একটি searching পদ্ধতি যাতে Data element এর পরিস্থিতির ভূমি ছাড়াই অপেক্ষাকৃত কর কর সময়ে search এবং Store করা যায়। এই পদ্ধতিতে key value এর মাধ্যমে জানিয়ে দেয় ডাটাবেজের কোন কোন Data store এবং কোন কোন থেকে Data retrieve হবে। Hash function Key value এর মাধ্যমে Direct address generate করে।

■ **B⁺ tree file organization:** B⁺ tree file organization হল indexed sequential access method এর advanced method। এটি ফাইলের সব রেকর্ড সংরক্ষণ করতে tree-like structure ব্যবহার করে। এটি key-index ধারণাটি ব্যবহার করে যেখানে রেকর্ডগুলি sort করার জন্য primary key ব্যবহৃত হয়। প্রতিটি primary key কোর্টের সাথে মাত্র index এর মান নির্ধারণ করা হয়। এবং রেকর্ডের সাথে মাত্র করা হয়। B⁺ tree, binary search tree (BST) এর অনুরূপ, তবে একটি দুটিরও বেশি children থাকতে পারে। এই

প্রতিটে, সমস্ত রেকর্ডগুলি কেবল leaf node এ সংরক্ষণ করা হয়। যাহাতো মোড়গুলি leaf node-র পফেলার হিসাবে কাজ করে। তারা কোন রেকর্ড ধারণ করে না।

■ **Clustred files organization :** দুই বা ততোধিক রেকর্ডকে একটি ফাইলে সংরক্ষণ করার প্রক্রিয়াকে ক্লাস্টার বলে। ফাইলগুলিতে একই তেটা ব্রকচিটে দুটি বা ততোধিক টেবিল থাকে এবং key attributes এর সাহায্যে ম্যাপিং করে ডাটা সংরক্ষণ করা হয়। ক্লাস্টারিং প্রতিটি ডাটা সংরক্ষণ এবং ডাটা Searching এ Efficiency বৃদ্ধি এবং complexity ক্রাস পায়। নিম্নের ক্লাস্টার এর উদয়ারণ দেওয়ো।

Employee

EmpID	Name	Address	DeptID
1	Jhon	Dhaka	11
2	Robart	Chattogram	10
3	Amy	Khulna	11
4	David	Rajshahi	12

Department

DeptID	DeptName
10	CSE
11	EEE
12	ME

Cluser Key

DeptID	DeptName	EmpID	Name	Address
10	CSE	2	Robart	Chattogram
11	EEE	1	Jhon	Dhaka
11		3	Amy	Khulna
12	ME	4	David	Rajshahi

Indexing Techniques

প্রশ্ন ১. ইনডেক্স কি? ইনডেক্স এর সুবিধা-অসুবিধা আলোচনা করুন। What is indexing? Discuss the advantages and disadvantages of indexing.**

উভয় ইনডেক্স - ডাটা ফাইলে ইনপুটকৃত ডাটাকে নির্দিষ্ট নির্যাম সাজানোর পদ্ধতিকে বলা হয় ইনডেক্স। কোন ফাইলের ডাটা ইনডেক্স করে অর্থ হচ্ছে সেই ফাইলের রেকর্ডগুলোকে উচ্চক্রমানুসারে (Ascending) বা নিম্নক্রমানুসারে (Descending) সাজানো। ডাটা ফাইল থেকে কোন নির্দিষ্ট ডাটাকে স্ক্রুত খুঁজে পাওয়ার জন্য ফাইলের প্রক্রসমূহকে ইনডেক্স করে রাখতে হয়।

প্রক্রস করার ন্যায় ইনডেক্স করে ডাটাবেজের রেকর্ডগুলিকে উচ্চ বা নিম্নক্রমানুসারে সাজানো যায়। প্রথমোւে ডাটাবেজের কোন রেকর্ড শেখেন বা সংযোজন করলে ইনডেক্স করা ফাইলেও তা আপডেট হয়। যাইহো ইনডেক্সের স্টেট চেয়ে স্ক্রুততর। সেজন্য বর্তমানে স্টেট না করে

ইনডেক্স করেই ডাটাবেজের রেকর্ডসমূহ উচ্চ বা নিম্নক্রমানুসারে সাজানো হয়।

ইনডেক্স-এর সুবিধাসমূহ : ইনডেক্স-এর মাধ্যমে রেকর্ড সাজানোর সুবিধাগুলো হলো :

- ১। ডাটা ফাইলকে ইনডেক্স করার পর নতুন রেকর্ড যুক্ত করা হলে তা নিজে থেকেই সাজানো হয়ে যায়।
- ২। ডাটা ফাইলকে ইনডেক্স করার পর একটি ইনডেক্স ফাইল তৈরি হয়।
- ৩। ডাটা ফাইলকে ইনডেক্স করার পর একটি ইনডেক্স ফাইল তৈরি হয়।
- ৪। একটিক ফাইলের মধ্যে সম্পর্ক তৈরি সহজ হয়।

ইনডেক্স-এর অসুবিধা : ইনডেক্স-এর মাধ্যমে রেকর্ড সাজানোর

- ১। ডাটা ফাইলকে ইনডেক্স করার পর একটি ইনডেক্স ফাইল তৈরি হয়ে যায়।
- ২। স্টেটেড ডাটাগুলো নিয়ে ইনডেক্স ফাইল তৈরি হয়।
- ৩। ইনডেক্স ফাইলকে সংরক্ষণ করার জন্য কম্পিউটারের স্মৃতিতে অতিরিক্ত জায়া লাগে।

প্রশ্ন ২. হ্যাশ টেবিল কিভাবে কাজ করে?

How Hash Table works:

একটি সাধারণ হ্যাশ টেবিল আইটেমগুলিকে শুধুমাত্র একটির list এর পরিবর্তে একটিক list এ রেখে কাজ করে।

প্রতিটি আইটেম কোন তালিকায় রাখতে হবে তা choose করতে এটি খুব দুটি এবং repeatable (পুনরাবৃত্তিযোগ্য) (অর্থাৎ non-random) পদ্ধতি ব্যবহার করে।

মুন্তরাং যখন আইটেমটি আবার খুঁজে বের করার সময় হয়, তখন কোন তালিকাটি দেখতে হবে তা discover করার জন্য এটি সেই পূর্বের পদ্ধতির পুনরাবৃত্তি করে এবং তারপর সেই তালিকায় একটি normal (slow) linear search করে।

NB: Data Structure এ বিজ্ঞারিত আলোচনা করা আছে।

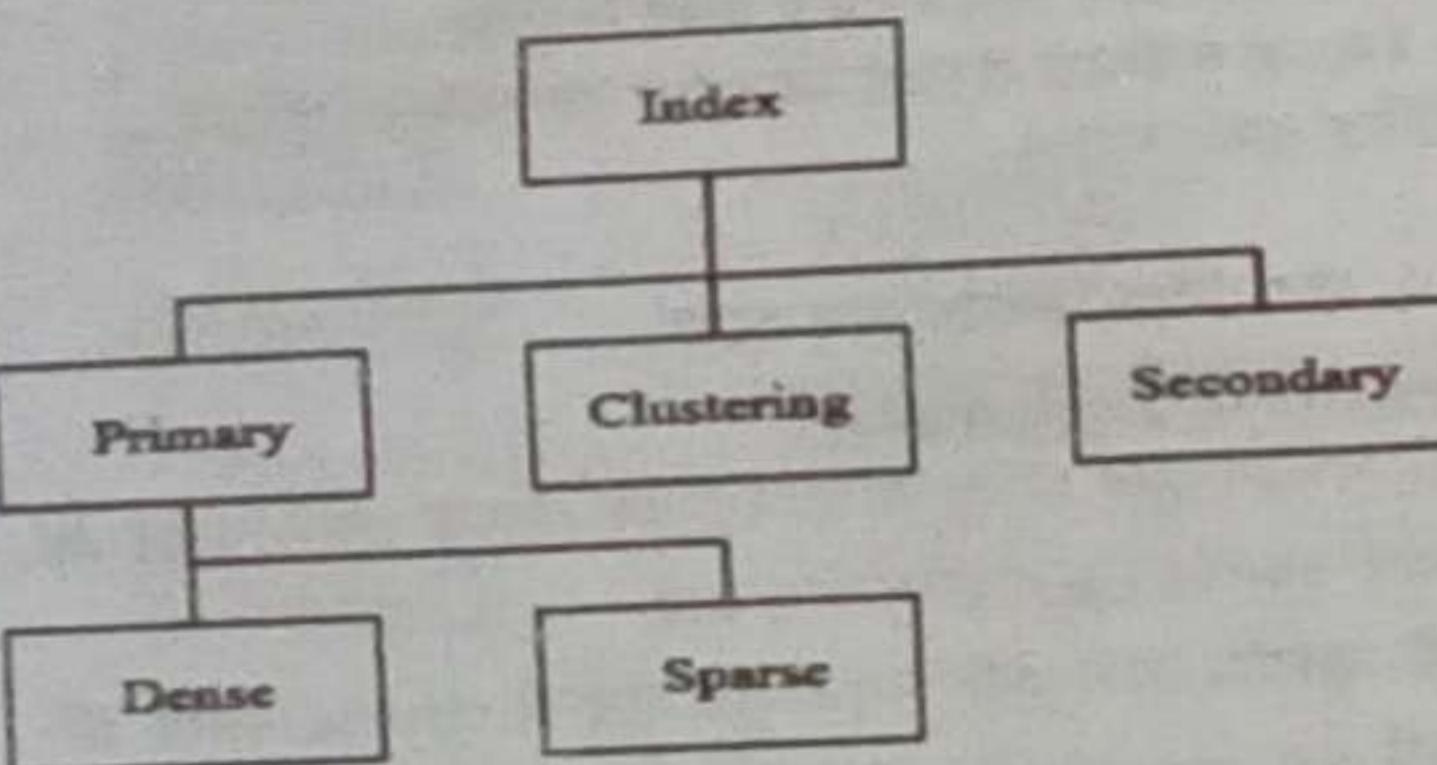
প্রশ্ন ৩. ইনডেক্স এবং স্ট্যট-এর মধ্যে পার্থক্য। Write the difference between indexing and sorting.**

উভয়ই ইনডেক্স এবং স্ট্যট-এর মধ্যে পার্থক্য।

ইনডেক্স	স্ট্যট
১। এলোমেলো ডাটাকে সুবিন্যস্তভাবে সাজানোর জন্য সময় ক্লুবন্মালকভাবে কর লাগে।	১। এলোমেলো ডাটাকে সুবিন্যস্তভাবে সাজানোর জন্য ইনডেক্স-এর চেয়ে সময় বেশি লাগে।
২। এই পদ্ধতিতে ডাটা ফাইলকে ইনডেক্স করার পর নতুন রেকর্ড যুক্ত করা হলে তা নিজে থেকেই স্ক্রুত হয়ে পাওয়ার জন্য ফাইলের প্রক্রসমূহকে ইনডেক্স করে রাখতে হয়।	২। এই পদ্ধতিতে ডাটা ফাইলকে স্ট করার পর নতুন রেকর্ড যুক্ত করা হলে নতুন করে আবার ফাইলটিকে স্ট করতে হয়।
৩। ইনডেক্স পদ্ধতিতে ডাটা ফাইলকে স্ট করা হলে মূল টেবিলে ফাইলের রেকর্ডের সিরিয়াল টিক থাকে না।	৩। স্ট্যট পদ্ধতিতে ডাটা ফাইলকে স্ট করা হলে মূল টেবিলে ফাইলের রেকর্ডের সিরিয়াল টিক থাকে না।

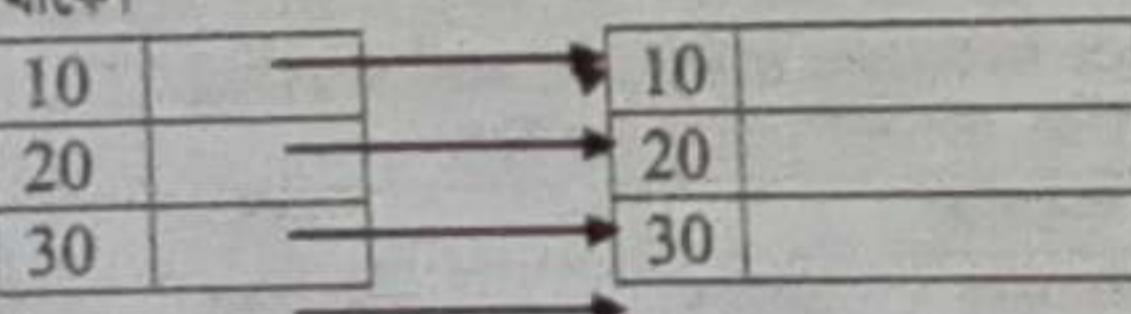
৪। ডাটা ফাইলকে ইনডেক্স করা হল নতুন ইনডেক্স ফাইল তৈরি হয় এবং মূল টেবিল ফাইল অবস্থায় স্থিত রয়ে থাকে।	৪। ডাটা ফাইলকে সঁজ করা হল মূল টেবিল ফাইলটিই সঁজে অবস্থায় স্থিত রয়ে থাকে।
৫। ইনডেক্স-এর মাধ্যমে সঁজ করা অতিরিক্ত ইনডেক্স ফাইল তৈরি হয় বলে অতিরিক্ত জারগার প্রয়োজন নেই।	৫। সঁজ-এর মাধ্যমে সঁজ করা হলে মূল টেবিল ফাইল পরিবর্তিত হয় বলে অতিরিক্ত জারগার প্রয়োজন নেই।

প্রশ্ন 8. ইনডেক্স-এর প্রকরণের আসন্নন করন।
উত্তর: Types of Indexing: Indexing attributes এর উপর ভিত্তি করে indexing কে সংজ্ঞাদিত করা হয়। Indexing can be of the following types.



a. Primary Index: যদি টেবিল এর index প্রাইমারি key এর উপর ভিত্তি করে বানানো হয় তখন তাকে Primary Indexing বলে।

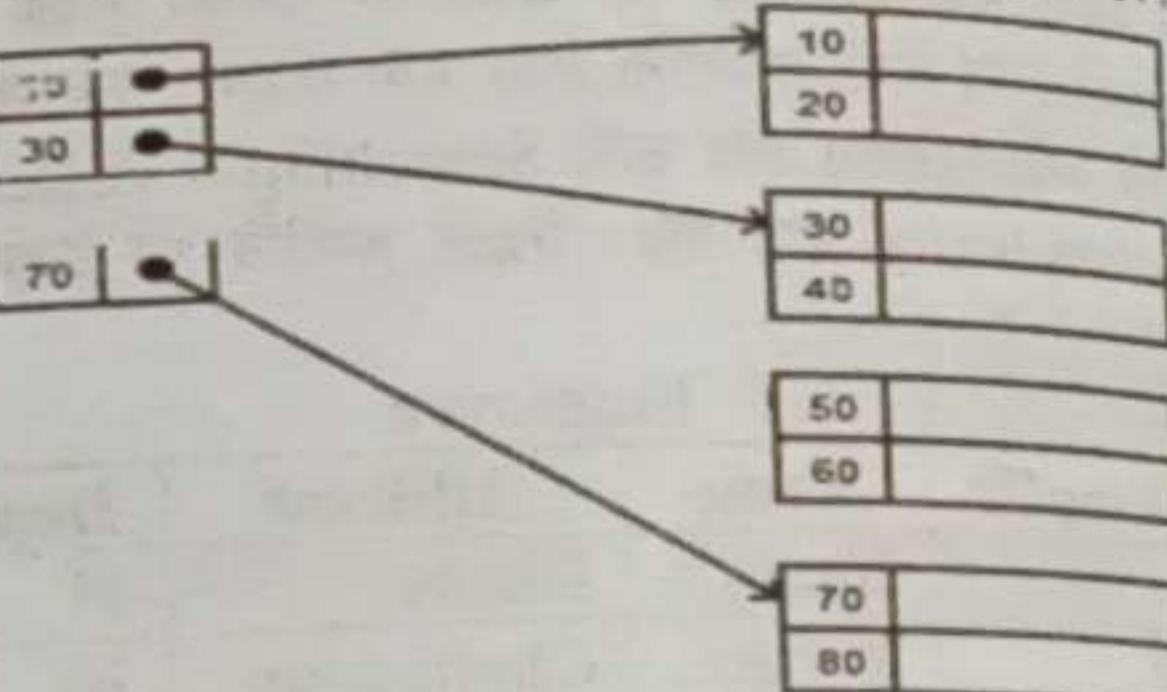
i. Dense index: dense index এ প্রতিটি search key value এর জন্য database এ একটি index record থাকে। এটি searching সূচ করে তোলে তবে যদিও index record সংরক্ষণ করার জন্য অতিরিক্ত store space প্রয়োজন। Index records search key এর value এবং তিনের actual record এর একটা pointer সংরক্ষণ করে থাকে।



প্রশ্ন c. Why do we use indexing in database? Justify whether or not indexing helps in query optimization process? [কেন আমরা ডাটাবেসে ইনডেক্স ব্যবহার করি? ইনডেক্সিং ক্ষেত্রের অস্তিমাইজেশন প্রক্রিয়া সাহায্য করে কিনা তা বিচার করুন।]

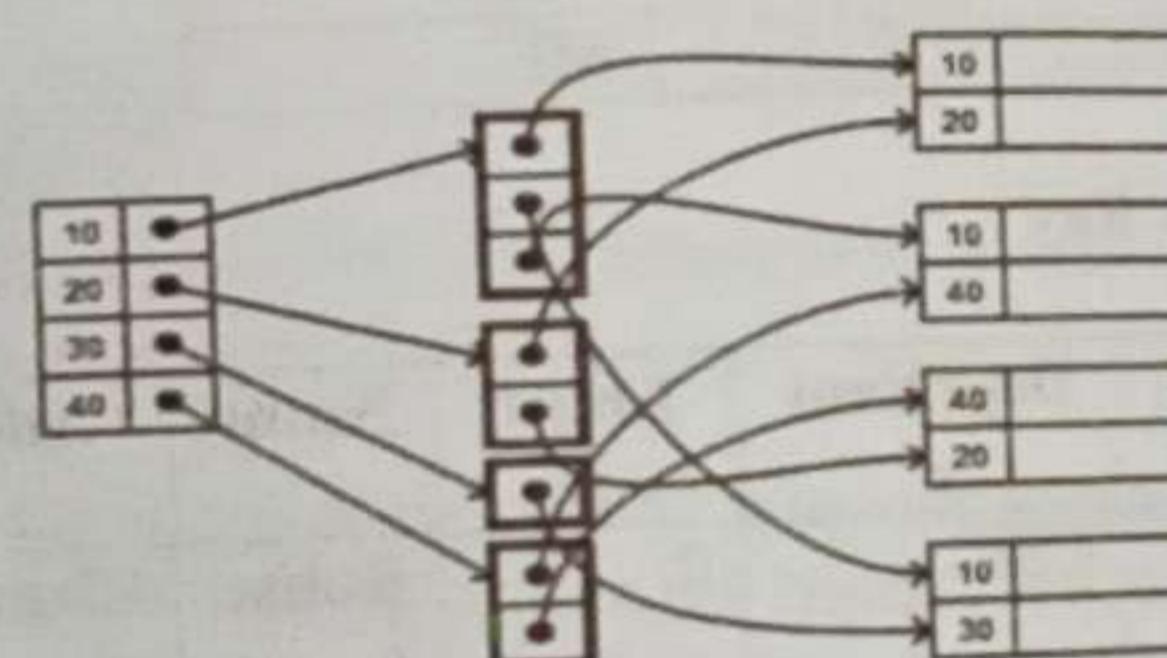
উত্তর: ডাটাবেসে ইনডেক্স ব্যবহার করা হয়, কারণ ডাটাবেসে প্রতিটি জো না খুজে সুলন্মূলক সহজ উপায়ে ইনডেক্সের মাধ্যমে নির্দিষ্ট ডাটা খুঁজে বাবে। ইনডেক্সিং কুরোর অস্তিমাইজেশন প্রসেস কে হেল্প করে। কোন ডাটা খুঁজে পেতে সহজেই পাওয়া যায়। নিম্নে একটা উদাহরণ মাধ্যমে বোঝার চেষ্টা করি। ইনডেক্সিং ছাড়া যদি zack নামে কাউকে কুরোর করা হয়েছে।

ii. Sparse Index: sparse index এ প্রতিটি search key value এর জন্য database এ index record থাকে না। Index records search key এর value এবং তিনের actual record এর একটা pointer সংরক্ষণ করে থাকে।



b. Secondary Index: Secondary index এ ধরনের file'de থেকে বানানো হয়, যেগুলো candidate key এবং নেকড গুলো থাকে অথবা non-key with duplicate values.

Note: Key field is the unique field in the table. Non-key field may not required to be unique.

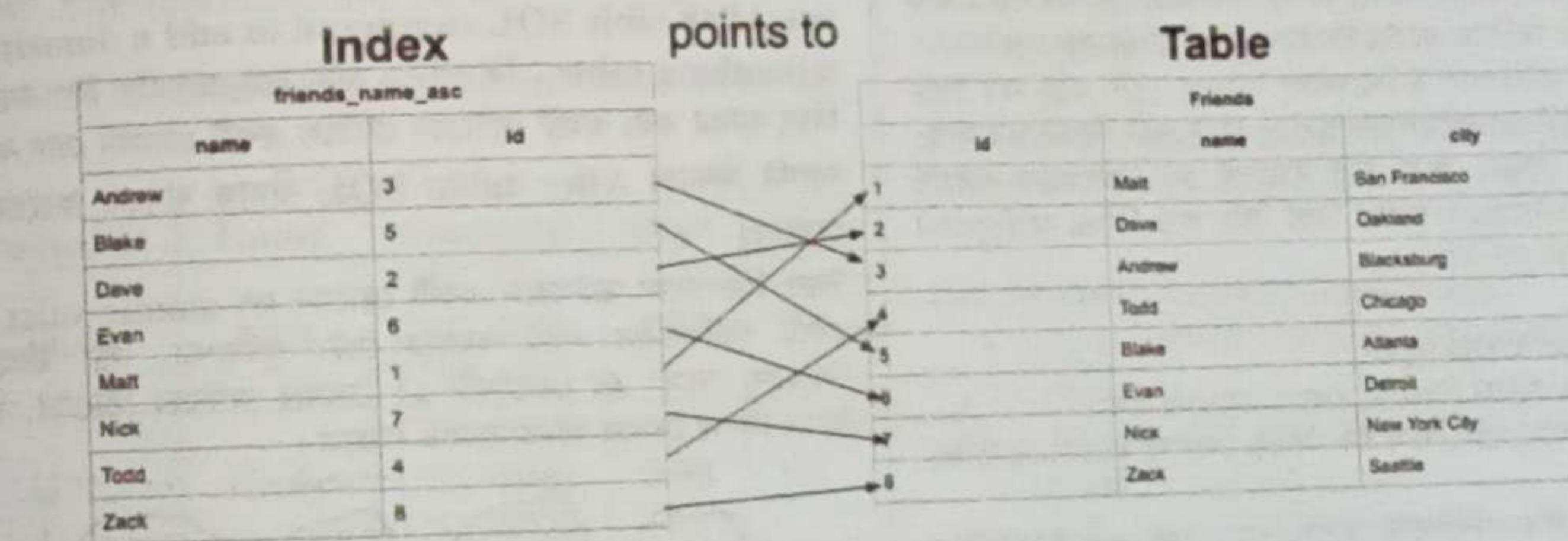


c. Cluster Index: clustering index হলো ordered data file যেখানে ডেটা ফাইলটি non-key field এ অর্ডার করা হয়।

SELECT * FROM friends WHERE name = 'Zack';

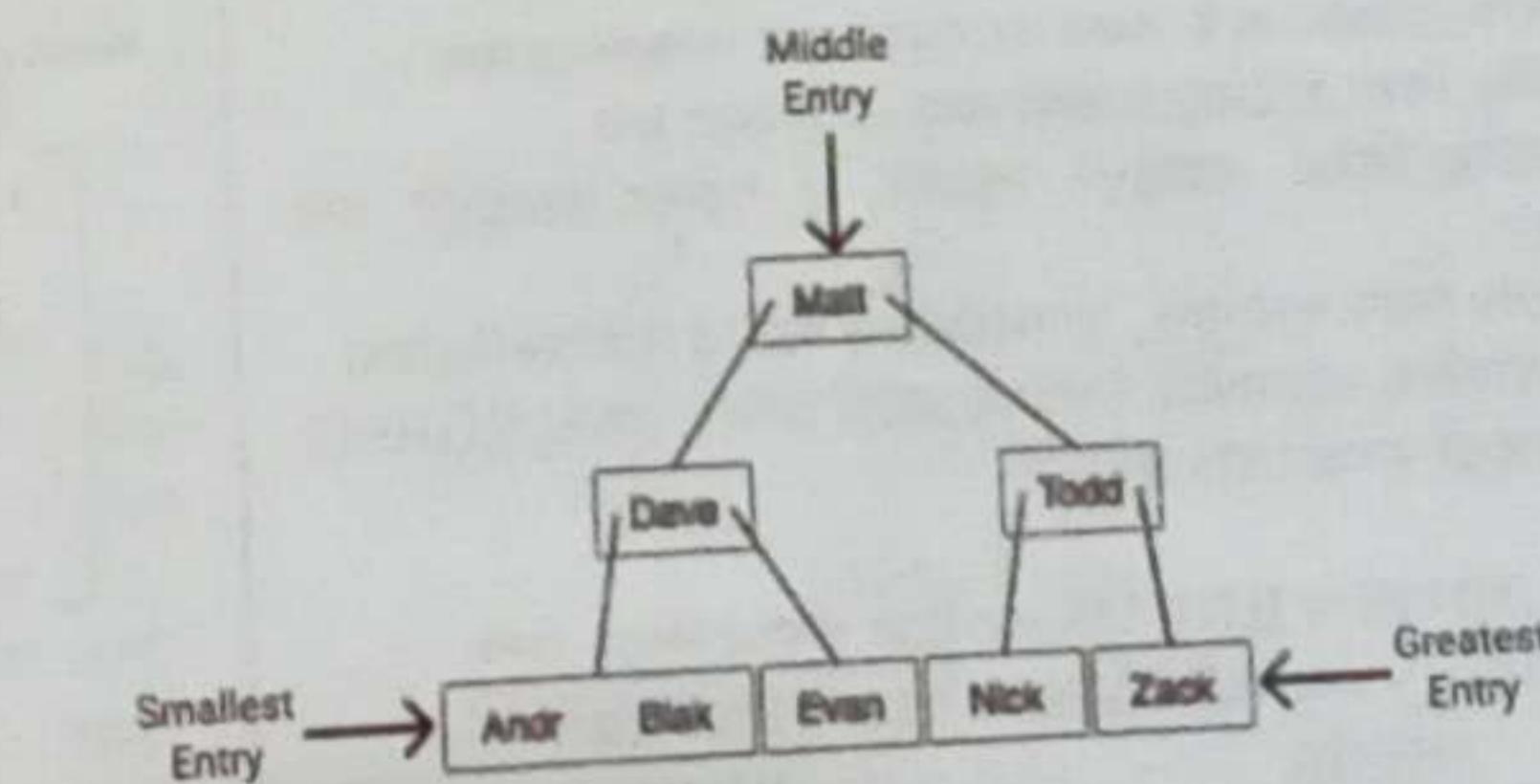
friends		
id	name	city
1	Matt	San Francisco
2	Dave	Oakland
3	Andrew	Blacksburg
4	Todd	Chicago
5	Blake	Atlanta
6	Evan	Detroit
7	Nick	New York City
8	Zack	Seattle

উপরের টিমে zack খুজতে প্রতিটা রো সার্চ করে মোট 8 বারে খুজে পাওয়া যাবে।



উপরের টিমে নামকে অ্যাসেন্ডিং ইনডেক্স ব্যবহার করা হয়েছে।

এখন পাশের টিমের মতো লজিকে মাত্র 2 বারেই zack কে খুজে পাবো। অথবা মিডল এন্ট্রি'তে আসবো, তারপর ছোট হলে বামে, বড় হলে তানে যাবো, বাইনারি সার্চ ইনডেক্সিং এর সাথে ব্যবহার করে কুরোর অস্তিমাইজেশন প্রসেসে হেল্প করে।



প্রশ্ন 6. Suppose, there is a table with following configuration: Animals= (AnimalGroup, AnimalName, AnimalAvgAge). Write a SQL statement to create an index on animal, group, and animal name.

উত্তর: Creating Clustered Indexes

The clustered index will be automatically created when the primary key is defined.
CREATE TABLE Animals (id INT PRIMARY KEY, AnimalGroup VARCHAR, AnimalName VARCHAR, AnimalAvgAge INT);

Non-Clustered Indexes

CREATE INDEX animal_name_asc ON Animals(AnimalName ASC);

ii) Relational Database Design

প্রশ্ন 1. Relational Database Management System কি? কয়েকটি RDBMS এর নাম লিখুন। ***
উত্তর: RDBMS এর পূর্ণাম Relational Database Management System। RDBMS সাধারণত ডেটাবেজ ও ডেটাবেজ ব্যবহারীর মধ্যে সমরকারী সফটওয়ার হিসেবে কাজ করে। এটি সাধারণত রিলেশনাল ডেটাবেজ মডেলের উপর ভিত্তি করে তৈরি করা হয়েছে। যেমন- Oracle, MySQL (Free software), Microsoft SQL Server, PostgreSQL (Free software), IBM DB2, Microsoft Access ইত্যাদি।

প্রশ্ন 2. RDBMS এর বৈশিষ্ট্য সমূহ এবং ব্যবহার লিখুন। ***

উত্তর: RDBMS এর বৈশিষ্ট্য সমূহ নিচেরূপ -

১। RDBMS এর সাহায্যে সহজেই টেবিল তৈরি ও ডেটা ইনপুট দেওয়া যাব এবং একাধিক টেবিলের মধ্যে রিলেশন তৈরি করা যাব।

২। RDBMS এর সাহায্যে একটি ডেটাবেজ একসাথে একাধিক ব্যবহারকারী ব্যবহার করতে পারে এবং এক ডেটাবেজের সাথে অন্য ডেটাবেজের তথ্য আদান-প্রদান করা যাব।

৩। RDBMS এর সাহায্যে ডেটাবেজ টেবিলের ডেটাগুলোকে সাজিয়ে উন্নয়ন করা যাব, ডেটা গুলোর ভিত্তি ফরমেটের রিপোর্ট তৈরি করা যাব এবং ডেটার ভিত্তিতে প্রযোজনীয় গ্রাফ ও চার্ট তৈরি করা যাব।

৪। উইকেডেজের একটি ক্লিয়েন্ট ইন্টারফেসে সাহায্যে ডেটা এক্সিফ কর্ম তৈরি করা যাব এবং ডেটাভাবে একটি ডেটাবেজ তৈরি করা যাব।

৫। অন্য কোন প্রয়োজন থেকে ডেটা ইমপোর্ট করে ডেটাবেজে ব্যবহার করা যাব এবং ডেটাবেজের ডেটার উপর ভিত্তি করে বিভিন্ন আপ্রস্তুকেশন সফটওয়্যারও তৈরি করা যাব।

■ RDBMS এর ব্যবহার সমূহ

১। বড় প্রতিষ্ঠানের বিভিন্ন বিভাগের তথ্য ব্যবহারণের জন্য।

২। অনলাইন বিমান, ট্রেন, বাস এবং বিভিন্ন খেলা বা ইভেন্টে এর টিকিট ব্যবহারণের জন্য।

৩। বাংকের ঘাবঠায় লেনদেনের হিসাব-নিকাশ রাখা এবং ইলেক্ট্রনিক পেমেন্ট সিস্টেম।

৪। ব্যবসা প্রতিষ্ঠান বা ই-কার্মাস এর ক্ষেত্রে তথ্য ব্যবহারণের জন্য।

৫। বিভিন্ন শিক্ষা প্রতিষ্ঠানের সকল তথ্য ব্যবহারণের জন্য।

৬। রাষ্ট্রীয় বিভিন্ন ভর্তৃপূর্ণ ম্যাগাজিন ও দণ্ডের ভর্তৃপূর্ণ তথ্য সংরক্ষণে।

৭। দেশের জনসংখ্যার তথ্য, আদমশুমারি ও জেটোর লিস্ট তৈরিসহ অনেক ক্ষেত্রে RDBMS এর ভর্তৃপূর্ণ ব্যবহার রয়েছে।

প্রশ্ন 3. DBMS ও RDBMS এর মধ্যে পার্ক্স লিখুন। ***

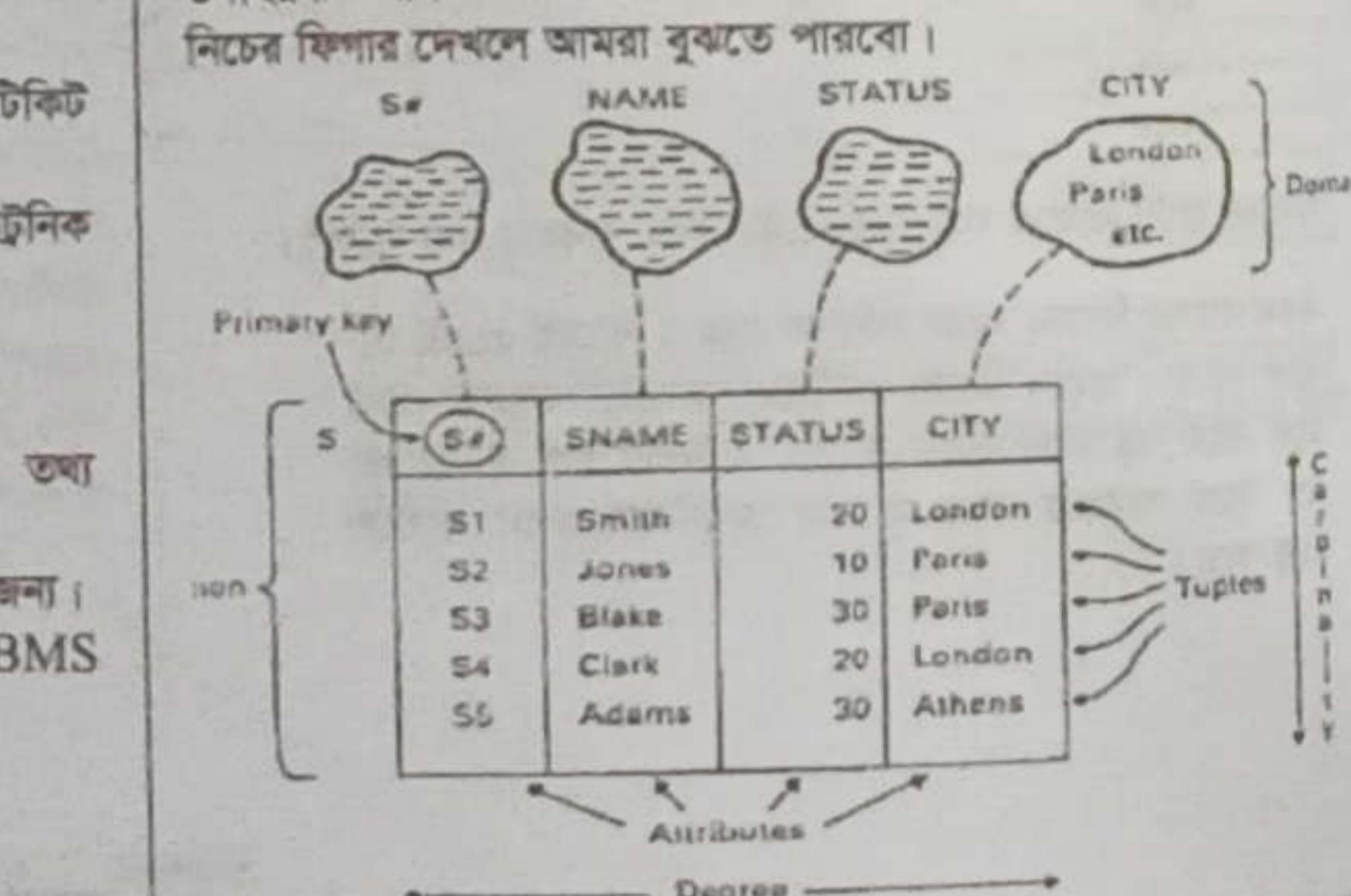
উত্তর: DBMS এবং RDBMS এর মধ্যে পার্ক্স নিচেরূপ -

DBMS	RDBMS
DBMS এর পূর্কল্প Database Management System.	RDBMS এর পূর্কল্প Relational Database Management System.
DBMS ডেটা ফাইল হিসেবে সংরক্ষণ করে।	RDBMS ডেটা টেবিল ফর্মে সংরক্ষণ করে।
ডেটা এলিমেন্ট এককভাবে আকস্মে করতে হয়।	একাধিক ডেটা এলিমেন্ট একসাথে আকস্মে করতে হয়।

ডেটা সমূহের মধ্যে কোন রিলেশন থাকে না।	ডেটা সমূহ টেবিলে সংরক্ষণ হয় এবং টেবিলগুলো একে অপরের সাথে সম্পর্কিত।
DBMS ডিফিনিউটেড ডেটাবেজ সাপোর্ট করে না।	RDBMS ডিফিনিউটেড ডেটাবেজ সাপোর্ট করে।
ডেটা রিভানডেপি থাকে।	ডেটা রিভানডেপি থাকে না।
অন্য সংখ্যাক ডেটা নিয়ে কাজ করার জন্য বিভিন্ন ছেট করার জন্য বিভিন্ন বিভিন্ন অর্গানাইজেশনে ব্যবহৃত হয়।	অধিক সংখ্যাক ডেটা নিয়ে কাজ করার জন্য বিভিন্ন ছেট করার জন্য বিভিন্ন বিভিন্ন অর্গানাইজেশনে ব্যবহৃত হয়।
DBMS সিস্টেম ইউজার সাপোর্ট করে।	RDBMS একাধিক ইউজার সাপোর্ট করে।
DBMS এ ডেটার লো-লেভেল সিকিউরিটি থাকে।	RDBMS এ ডেটার মাই- লেভেল সিকিউরিটি থাকে।
উদাহরণ- Microsoft Access ইত্যাদি।	উদাহরণ- XML, MySQL, PostgreSQL, SQL Server, Oracle, ইত্যাদি।

প্রশ্ন 8. What is domain in a relational database?
Explain with an example. Show how you would use Alter table SQL command to add a domain on a database table. [রিলেশনাল ডাটাবেসে ডোমেইন কি? উদাহরণ দিয়ে ব্যাখ্যা কর। একটি ডাটাবেস টেবিলে একটি ডোমেন যোগ করতে আপনি কিভাবে Alter table SQL কমান্ড ব্যবহার করবেন তা দেখান?]

উত্তর: রিলেশনাল ডাটাবেসে একটি ডোমেন হল atomic value এর একটি পূর্ণবৰ্ধারিত একই ধরণের সেট: পূর্ণসংখ্যা, স্ট্রিং ইত্যাদি। উদাহরণ: "নাম" এর ডোমেইনটি ১২ দৈর্ঘ্যের অক্ষরের স্ট্রিংের সেট। নিচের ফিল্ডের দেখলে আমরা বুঝতে পারবো।



Alter Table: To add a new column 'email' at the end of the table 'agent1' with field name and data type:

Field Name	Data Type	Size	Decimal Places	NULL	Constraint
e-mail	char	25		No	

the following SQL statement can be used:
**ALTER TABLE agent1
ADD email char (25);**

Normalization

প্রশ্ন 1. ডাটাবেজ নরমালাইজেশন সর্বোকে বিশদ বর্ণনা করুন। ***
Normalization কি? প্রযোজনীয়তা সেৰি। [PSC(AP)-2019, 36 BCS]

উত্তর: Normalization - ডাটাবেস ম্যানেজমেন্ট সিস্টেম এর রিলেশনাল ডাটাবেস এর redundancy ও Dependency কে Minimize করার জন্য এর Field ও Table সমূহকে সুসংগঠিত করার ক্ষেত্রে যে পক্ষতি ব্যবহৃত হয় তাকে Normalization বলে। এর উদ্দেশ্য হল যাতে ডাটা রিভানডেপি (data redundancy) না থাকে এবং ডাটা ইন্টিগ্রিটি (data integrity) বজায় থাকে।

Database Normal Forms [36 BCS, PSC(ANE)-2020]

- 1NF (First Normal Form)
- 2NF (Second Normal Form)
- 3NF (Third Normal Form)
- BCNF (Boyce-Codd Normal Form)
- 4NF (Fourth Normal Form)
- 5NF (Fifth Normal Form)
- 6NF (Sixth Normal Form)

এখন Student টেবিলের সব কলামের ভ্যালু একক (atomic) হয়েছে। আবার আমরা দেখতে পাচ্ছি যে, একটি ডাটা বার বার আসছে। তবুও CourseID কলামের ডাটা পরিবর্তন হচ্ছে।

■ Second normal form (2NF): Second normal form (2NF) এ আমদের Student টেবিলটিকে দিয়ে যাব।

Table: Student

StudentID	Name	CourseID	Dept	dID
134001	Alex	CSE-3311	CSE	C4
134001	Alex	CSE-4623	CSE	C4
134001	Alex	CSE-4721	CSE	C4
132002	Healy	CSE-3311	EEE	E2
132002	Healy	CSE-4623	EEE	E2
132002	Healy	CSE-4721	EEE	E2
131003	Paterson	CSE-3311	CE	C1
131003	Paterson	CSE-4623	CE	C1
131003	Paterson	CSE-4721	CE	C1

উপরের টেবিলটিতে কোনো নরমালাইজেশন নেই। এই টেবিলে নতুন ডাটা যোগ করতে (data insert) পূর্বে ডাটা পরিবর্তন (data update) করতে এবং ডাটা মুছে ফেলতে (data delete) আমদের কিছু অসুবিধা হবে। (সেগুলোকে নরমালাইজেশন দ্বারা দূর করা যাব।) যেমন, আমরা যদি নতুন একজন Student এই টেবিলে যোগ করতে চাই যে কোনো সাবজেক্টই নেয় নি, তাহলে CourseID কলামে NULL ভ্যালু যাব। আবার আমরা যদি একজন স্টুডেন্ট এর সাবজেক্ট বাঢ়াতে বা কমাতে চাই তাহলে আমরা যুব সহজে তা করতে পারব না, কারণ CourseID কলামে ডাটা কমা দিয়ে আলাদা করা আছে।

■ First normal form (1NF): First normal form (1NF) এর শর্ত হচ্ছে টেবিলের সব কলামের ভ্যালু একক (atomic) হতে হব। আমরা দেখতে পাচ্ছি যে CourseID কলামের ডাটা একক (atomic) নয়। নিচে দেখানো উপায়ে আমরা Student টেবিলটিকে পরিবর্তন করে First normal form (1NF) গঠন করা হল।

1NF (First Normal Form) Rules:

- প্রতিটি টেবিল ঘরে একটি মান থাকতে হবে। [Each table cell should contain a single value.]
- প্রতিটি রেকর্ড unique হতে হবে। [Each record needs to be unique.]

Table: Student

StudentID	Name	CourseID	Dept	dID

<tbl_r cells="5" ix="4

[candidate key] থানে এমন কলাম বা কলামের সমষ্টি যা একটি টেবিলের প্রতিটি রেকর্ড কে আলাদা ভাবে চিহ্নিত করতে পারে। একটি টেবিলের এক বা একাধিক candidate key থাকতে পারে। এর মধ্যে একটি বিশেষ candidate key কে আমরা primary key বলি। যে attribute/column কোনও candidate key এর অংশ নয় তাকে non-prime attribute অথবা non-key attribute বলে।

আমদের Student টেবিলটি First normal form (1NF) এ আছে। আমদের বিভিন্ন শর্তটি পূরণ করতে হবে। Student টেবিল থেকে আমরা লক করি যে, [StudentID, CourseID] কলাম দুটি মিলে হচ্ছে একটা candidate key এবং name, Dept কলামগুলো হচ্ছে non-prime attribute।

এখন name, Dept কলামগুলি তথ্যাত্মক StudentID কলামের উপর নির্ভরশীল এবং StudentID হল candidate key: [StudentID, CourseID] এর একটি subset।

আমরা Student টেবিলটিকে উপরের ঘট করে Second normal form (2NF) এ নিতে পারি। আমরা একটি নতুন টেবিল Course তৈরি করলাম Student এবং Course এর মধ্যে সম্পর্ক ঠিক রাখার জন্য। তাহলে আমদের ডাটাবেজ এখন Second normal form (2NF)-এ চলে আসল।

■ **Third normal form (3NF):** Third normal form (3NF) এ নেওয়ার জন্য আমদের নিচের দুটি শর্ত পূরণ করতে হবে -

3NF (Third Normal Form) Rules

- Rule 1- অবশ্যই বিভিন্ন নরমাল ফর্ম এ থাকতে হবে। [Be in 2NF]
- Rule 2- কোন ট্রান্সিটিভ ফাংশনাল ডিপেনডেন্সি থাকবে না। [Has no transitive functional dependencies]

■ **Transitive functional dependency :** মনে করি একটি টেবিলের প্রাইমারি কি (primary key) হচ্ছে A এবং এই টেবিলের দুটি নন-প্রাইম (non-prime) কলাম হচ্ছে B এবং C, যেখানে C কলামের ভ্যালু যতটা A কলামের ভ্যালুর উপরে নির্ভরশীল তার চাইতে B কলামের ভ্যালুর উপরে বেশি নির্ভরশীল, আবার B কলামের ভ্যালু A কলামের ভ্যালুর উপরে সরাসরি নির্ভরশীল, তাহলে আমরা বলতে পারি যে C কলাম transitivity কলাম A এর উপর নির্ভরশীল।

আমদের Student টেবিলটিতে StudentID হচ্ছে প্রাইমারি কি (primary key) এবং name, Dept , did হচ্ছে নন-প্রাইম (non-prime) কলাম। আমরা লক করি যে, Dept কলামটি যতটা StudentID কলামের উপরে নির্ভরশীল তার চাইতে বেশি নির্ভরশীল DeptID কলামটির উপরে এবং DeptID কলাম StudentID উপরে সরাসরি নির্ভরশীল। সুতরাং আমরা বলতে পারি যে Dept কলামটি transitivity কলাম StudentID এর উপর নির্ভরশীল।

তাই Student টেবিলটিকে Third normal form (3NF)-এ নিতে নিচের মতো পরিবর্তন করতে পারি এবং Department নামে একটি নতুন টেবিল তৈরি করতে পারি।

Table: Student

StudentID	Name	dID
134001	Alex	C4
134002	Healy	E2
134003	Paterson	C1

Table: Course

StudentID	CourseID
134001	CSE-3311
134001	CSE-4623
134001	CSE-4721
132002	CSE-3311
132002	CSE-4623
132002	CSE-4721
131003	CSE-3311
131003	CSE-4623
131003	CSE-4721

Table: Department

dID	Dept
C4	CSE
E2	EEE
C1	CE

■ **Boyce and Codd Normal Form (BCNF):** এটি Third Normal form এর higher version। 3NF থারে সব ব্যক্তিগত handle করা যায় না সেই ব্যক্তিগত গুলো deal করার জন্য BCNF ব্যবহার করা হয়। একটি 3NF table যেটির multiple overlapping candidate keys নেই তাকে BCNF বলে। একটি table কে BCNF বলা হবে যদি নিচের শর্ত পূরণ করে।

- 3rd Normal Form এ থাকতে হবে।
- এবং, প্রতিটি functional dependency ($X \rightarrow Y$) এর ক্ষেত্রে X কে অবশ্যই super Key হতে হবে।

Fourth Normal Form (4NF): একটি table কে Fourth Normal Form বলা হবে যদি নিচের শর্ত পূরণ করে।

- Boyce-Codd Normal Form এ থাকতে হবে।
- এবং, কোন Multi-Valued Dependency থাকা যাবে না।

5NF (Fifth Normal Form): একটি টেবিলকে 5th নরমাল ফর্ম বলা যাবে তখনই-

- যখন টেবিল 4NF এ থাকে
- এবং টেবিল কোন প্রকার ডাটা লস না করে যে কোন সংখ্যাক ছোট টেবিলে decomposed করা যাবে না।

6NF (Sixth Normal Form) Proposed

6th Normal Form is not standardized, তবুও এটি কিছু সময়ের জন্য ডাটাবেস বিশেষজ্ঞদের দ্বারা আলোচনা করা হচ্ছে। আপ

করি, অন্দর ভবিষ্যতে আমদের 6th Normal Form এর জন্য একটি clear & standardized definition থাকবে।

Normalization এর প্রয়োজনীয়তা - যাতে ডাটা রিভালডেলি না থাকে এবং ডাটা integrity বজায় থাকে। ডাটাবেজ ডিজাইনের ক্ষেত্রে সব দুটা যায় একটি ডাটা বারবার বিভিন্ন টেবিলে এমনভাবে আসে, যেখনে টেবিলগুলো একটু অন্যভাবে ডিজাইন করলেই অনেক ডাটা বেঁচে যাবে। ডাটাবেজ নরমালাইজ করার মাধ্যমে আমরা নিচিত করি যে ডাটাবেজে রিভালডেলি ডাটা থাকবে না।

ডাটা ইনটিগ্রিটি Integrity শব্দের অর্থ হচ্ছে উক্ততা। অনেক সময় ডাটাবেজে বিভিন্ন কারণে (হার্ডওয়ারের ঝুঁটি কিংবা সফটওয়ারের সমস্যা বা ডাটাবেজ ডিজাইনের সমস্যা) ডাটায় ভেজাল চুক্কে যায়। নরমালাইজেশন করলে ডাটার উক্ততা বজায় থাকার সম্ভাবনা বেড়ে যায় অনেক।

প্রশ্ন 2. What is Partial Dependency?

যখন একটি non-prime attribute কোন candidate key এর প্রের functionally dependent হয় তখন তাকে Partial Dependency বলে।

Example: StudentProject

StudentID	ProjectNo	StudentName	ProjectName
S01	199	Katie	Geo Location
S02	120	Ollie	Cluster Exploration

উপরের table টিকে partial dependency আছে; কিভাবে আছে চলুন দেখে নেয়া যাক-

StudentID = Unique ID of the student

StudentName = Name of the student

ProjectNo = Unique ID of the project

ProjectName = Name of the project

StudentID এবং Project No হল prime key attributes।

StudentName এবং ProjectName হল non-prime key attributes যাদেরকে Partial Dependent হতে হলে candidate key এর উপর functionally dependent হতে হবে।

StudentName, ProjectName কে StudentID এবং ProjectNo সনাক্ত করা যায়, তাই বলা যায় এই relation টিকে Partial dependency আছে।

মুকারং "StudentProject" violate করছে 2NF, Partial Dependency remove করতে table টিকে decompose করতে হবে নিচের মত করে-

Exercise

Normalization উদাহরণ:

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3 rd Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.
Robert Phil	5 th Avenue	Clash of the Titans	Mr.

Table 1

এখনে Movies Rented কলামে আমরা একাধিক ভ্যালু দেখতে পায়। তাই এখন 1st Normal Form এ ঘাইতে হবে (সিংগ্ল ভ্যালু করতে হবে)।

উপরের টেবিলটি 1NF এ এমন দেখতে হবে

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3 rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3 rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5 th Avenue	Clash of the Titans	Mr.

Table 1: In 1NF Form

আমরা এখন 2NF এ চলে যাব (Partial Dependency দূর করবো)। আমরা উপরের সারণিকে বিভাজন না করা পর্যন্ত আমদের ডাটাবেসটিকে 2nd Normalization ফর্ম হিসাবে তৈরি করতে পারি না। তাই আমরা উপরের টেবিলকে নিচের মত বিভাজন করবো।

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3 rd Street 34	Mr.
3	Robert Phil	5 th Avenue	Mr.

Table 1

টেবিল ১,২ রেকর্ড এর primary key ব্যবহার করে table ১,২ এর
রেকর্ডকে সংজ্ঞাবে চিহ্নিত করতে পারে :

Let's move into 3NF: আমদের আবার আমদের 2NF
টেবিলটি বিভক্ত করা দরকার।

3NF Example

MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION ID
1	Janet Jones	First Street Plot No-4	2
2	Robert Phil	3rd Street 34	1
3	Robert Phil	5th Avenue	1

TABLE 1

MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshall
2	Daddy's Little Girls
3	Clash of the Titans

Table 2

SALUTATION ID	SALUTATION
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

Table 3

আমদা আবার আমদের টেবিলগুলিকে বিভক্ত করেছি এবং একটি নতুন টেবিল তৈরি করেছি যা Salutations সংরক্ষণ করে। কোনও transitive functional dependencies, নেই এবং তাই আমদের টেবিলটি 3NF এ রয়েছে। Table ৩ এ Salutation ID primary key, এবং Table 1 এ Salutation ID টি foreign key, এখন আমদের ছোট উন্নয়নগুলি এমন লেভেলে রয়েছে যা আরও normalization এর উচ্চতর ফর্মালি অর্জন করতে পারে না। প্রকৃতপক্ষে, এটি ইতিমধ্যে higher normalization forms এ রয়েছে।

প্রশ্ন ৩. Why 1NF, 2NF, 3NF and BCNF are implemented?

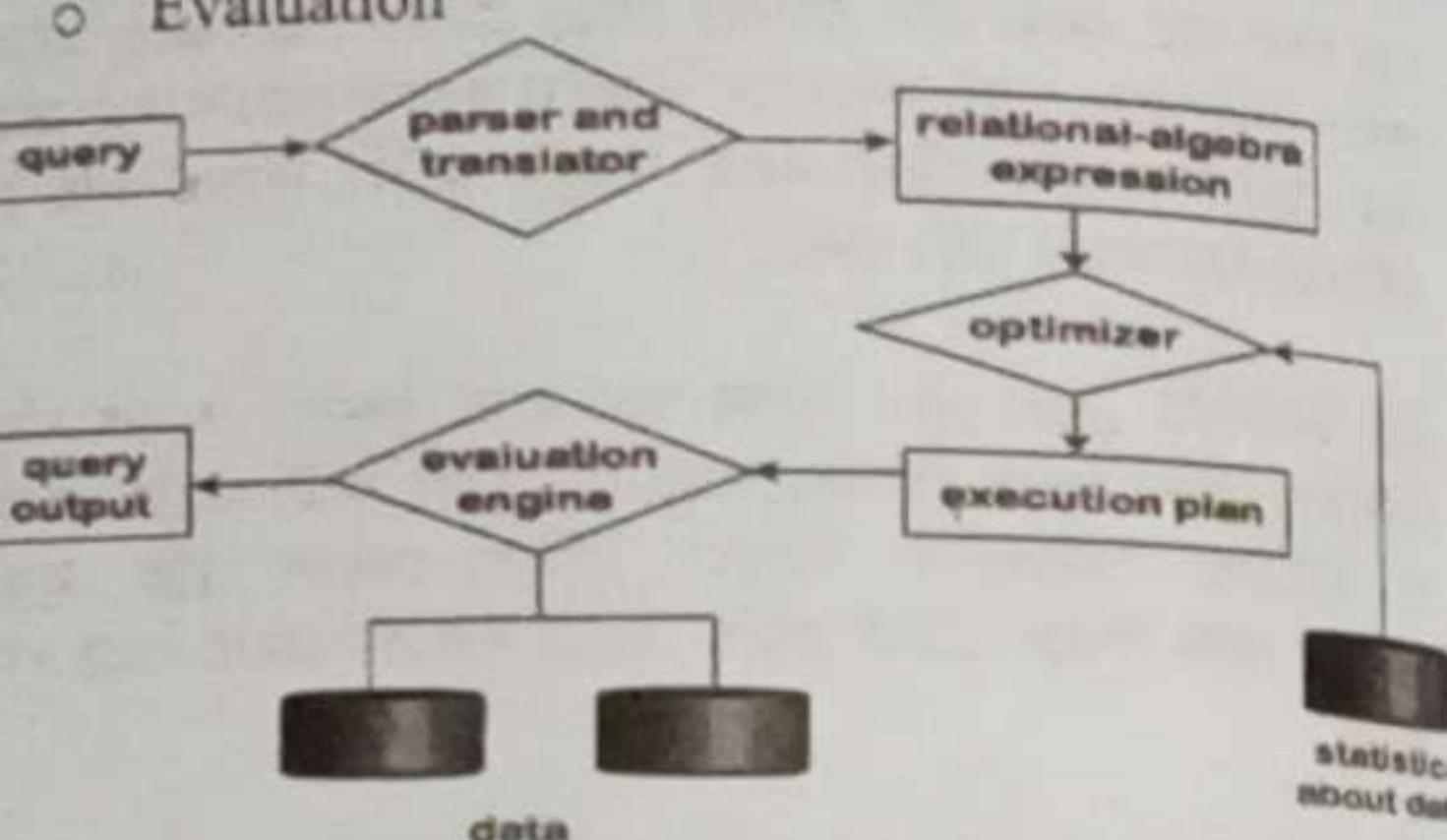
উত্তর: 1NF, 2NF, 3NF এবং BCNF implemented করা হয় মূলত বিভিন্ন টেবিলগুলি ফর্মেটে ভেটা রিভানডেলি (পুনরাবৃত্তি) এবং Insertion, Update এবং Deletion Anomaly এর মতো অনাকস্তিক বৈশিষ্ট্যগুলি অপসারণের জন্য টেবিলে decompose করে। এটি একটি ধারাবাহিক multi-step অক্ষিয়া যত সত্ত্ব তাটা রিভানডেলি দূর করে এবং relation টেবিলগুলি থেকে সন্দৰ্শ তো সরিয়ে ভেটাকে টেবিলগুলির আকারে রাখে।

Query Processing

কুয়েরি প্রসেসিং - কুয়েরি প্রসেসিং high level query কে low level expression এর পর্যালোচনা করে। query optimization এবং executed query এর সঠিক ফলাফলের জন্য কুয়েরি প্রসেসিং ব্যবহার করা হয়।

কুয়েরি প্রসেসিং এর ধৰ্ম সমূহ :

- Parsing and translation
- Optimization
- Evaluation



Steps in query processing

Parsing and translation : Parser User এর execute SQL command থেকে syntax analysis, relation analysis, tuple এবং attribute থেকে syntax tree তৈরি করে যা 'parse-tree' নামে পরিচিত। Parser parse-tree থেকে SQL command কে রিলেশনাল Algebra এ রূপাল্পন করে এবং translator রিলেশনাল Algebra কে query system এর internal format এ রূপাল্পন করে। ধরন আপনি টেবিল থেকে এই সম্পর্ক employee দের খুজে বের করতে চান যাদের salary > 10000 এর বেশি। SQL command

SELECT emp_name
FROM Employee
WHERE salary>10000;

Parsing এবং translation human readable SQL command কে প্রথমে রিলেশনাল Algebra এবং রিলেশনাল Algebra query system এর internal format এ রূপাল্পন করে আউটপুট প্রদান করে।

Relational Algebra

$\sigma_{\text{salary} > 10000}(\pi_{\text{salary}}(\text{Employee}))$
 $\pi_{\text{salary}}(\sigma_{\text{salary} > 10000}(\text{Employee}))$

Optimization : ডাটাবেস সিস্টেমের minimum cost এfficient query evaluation এর plan রয়েছে, যা query optimizer নামে পরিচিত। Query optimizer প্রতিটি অপারেশনের time এবং space complexity analysis করে সর্বচেয়ে lowest cost path নির্ধারণ করে আউটপুট প্রদান করে।

Evaluation : User এর প্রদত্ত SQL command এর নির্ভরযোগ্য আউটপুট এর জন্য কোন ইনডেক্সিং এবং কোন অ্যালগরিদম ব্যবহার করে আউটপুট প্রদান করবে তা Evaluation plan নির্ধারণ করে।

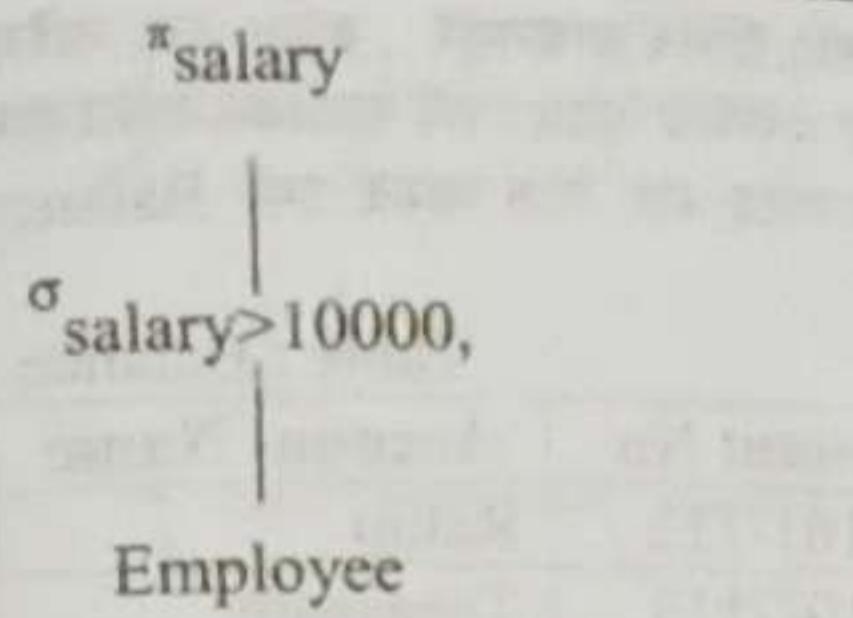
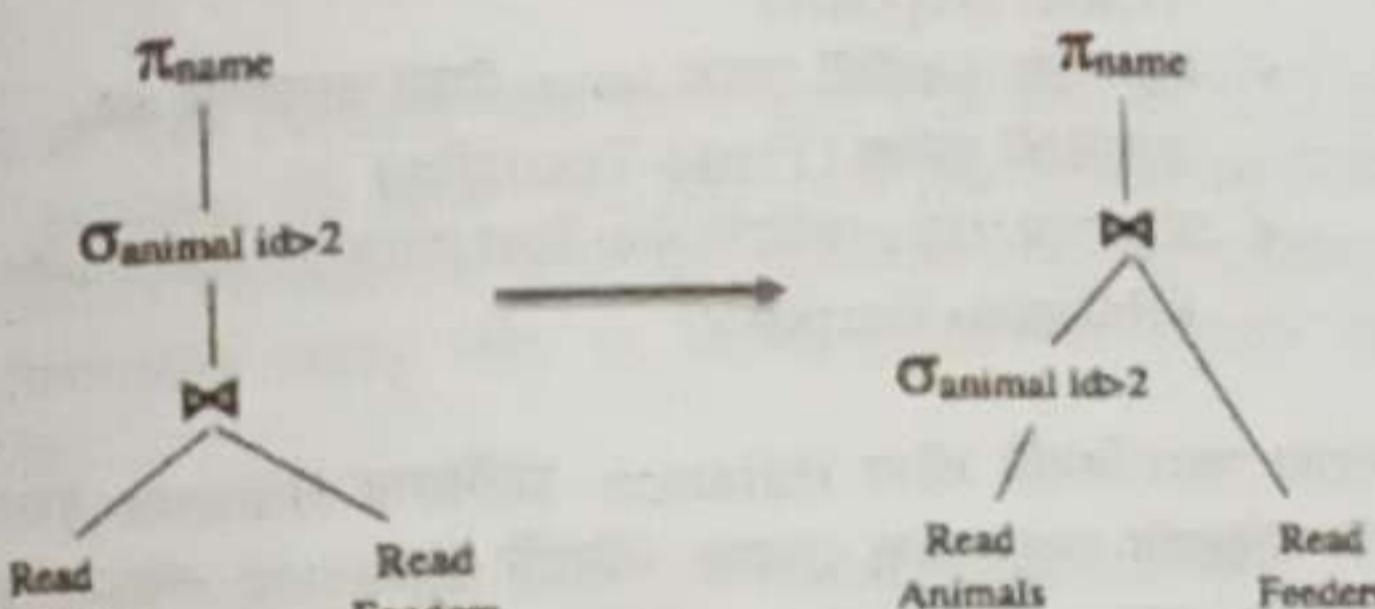


Fig: Query evaluation plan

Example-Query Processing: 1. "Select animals' names and their feeders from Dataset 'animals' where animal id>2" (before and after optimization)



Note: Symbol Denotes,

- Selection σ
- Projection π
- Join \bowtie

Join Operation

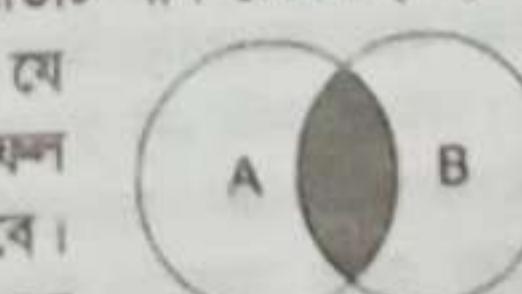
প্রশ্ন ১. Join operation কি? উদ্দয়ন সহ Join operation সম্বন্ধে বিশদ বর্ণনা করুন। [PSC(AP)-2019]

উত্তর: **Join operation** – প্রয়োজ্য column যুক্ত এবং উপর চিহ্ন করে, দুই বা ততোধিক table এর row গুলোকে combine করার জন্য JOIN clause ব্যবহার করা হয়।

Join operation ২ ধরনের : যথা:

- a) Inner Join
- b) Outer Join

■ **Inner Join:** Inner Join কীওয়াড়টি বাম টেবিল (A) এর সম্পর্ক রেকর্ড এর সাথে ডান টেবিল (B) এর যে সকল রেকর্ডগুলির মিল থাকবে সেই সকল ফলাফল যাম টেবিল (table1) থেকে প্রদর্শিত হবে। ইন্নার জয়েনকে $(A \cap B)$ Denote দ্বারা করা হয়।



Example:

SELECT Table_A.column1, Table_B.column2...
FROM Table_A A
INNER JOIN Table_B B
ON A.Key = B.Key

■ **Outer join operation:** Outer join operation হল join operation এর Extended ফিল। Outer join operation ৩ ধরনের : যথা:

- Left outer join operation
- Right outer join operation
- Full outer join operation

a) **Left Outer join:** LEFT JOIN কীওয়াড়টি বাম টেবিল

(table1) থেকে সম্পর্ক রেকর্ড

এর সাথে ডান টেবিল (table2)

এর রেকর্ডগুলির যদি কোন মিল

না থাকে তাহলে ডান নিকেল

table এর ফলাফল গুলো

NULL হয়। কিছু ডাটাবেস left join কে left outer join বলে।

Customers

CustomerID	Name	Address
1	Alfreds Futterkiste	USA
2	Richter	UK
3	Rose	AU

OrderID	OrderDetails	CustomerID
1	iPhone X	1
2	Samsung S10	2

SELECT Customers.CustomerName,

Orders.OrderID

FROM Customers

LEFT JOIN Orders ON Customers.CustomerID =

Orders.CustomerID

ORDER BY Customers.CustomerID;

Result:

Cutomer Name	Customer ID
Alfreds Futterkiste	1
Richter	2
Rose	Null

b) **Right outer join:** RIGHT JOIN কীওয়াড়টি বাম টেবিল

(table1) থেকে সম্পর্ক রেকর্ড এর সাথে ডান টেবিল (table2)

এর রেকর্ডগুলির যদি কোন মিল

না থাকে তাহলে বাম নিকেল

ফলাফল গুলো

NULL হয়, যেখা থাকলে বাদ যায়।

কিছু ডাটাবেস Right join কে Right outer join বলে।

এখন তার জন্য ট্রানজেকশন A কর হল। ট্রানজেকশন A প্রথমে সেবারে যে এই পদ্ধতির ২৮০ টি স্টকে আছে কিনা। ধরে নেই সাইটের ভাটাবেজে Product_Inventory নামে একটি টেবিল আছে যাতে পদ্ধতির পরিমাণ আছে। তো ট্রানজেকশন A সেই Product_Inventory টেবিল থেকে পেল যে তার ইউজার যে প্ল্যাটি কিনতে চায় তা ৫০০ টি আছে। সুতরাং ট্রানজেকশন A এই অর্ডারটিকে কনকুর্স করল এবং Product_Inventory টেবিলে এই প্ল্যাটির পরিমাণ ৫০০ থেকে কমিয়ে ২২০ করে দিল। কিন্তু ট্রানজেকশন A এখনও কর্মসূচি করত হচ্ছে। একই সময়ে অবৈধ একজন একই সাইট থেকে ঠিক এই প্ল্যাটি ৪০০ টি কিনতে চাইল। ধরে নেই পদ্ধতির ক্ষেত্রে জন্য ট্রানজেকশন B কর হল। ট্রানজেকশন B সেবারে যে এই প্ল্যাটি মাত্র ২২০ টি আছে, তাই সে ক্ষেত্রে জানাল যে তার অর্ডারটি নেয়া যাচ্ছে না। এরই মধ্যে আবার ট্রানজেকশন A এর যে ক্ষেত্রে সে অর্ডারটি বাতিল করে দিল, তার ফলে ট্রানজেকশন A রোলব্যাক হতে পেল। অর্থাৎ Product_Inventory টেবিলে এই প্ল্যাটির পরিমাণ আবার ৫০০ হয়ে পেল। তার মানে এখনে ট্রানজেকশন B এখন একটি ভাটা পেয়েছিল যা আসলে কর্মসূচি হচ্ছি আর এটাকেই Dirty Read বলে।

Non Repeatable Read – একটি ট্রানজেকশন যদি একই ভাটা দুবার পড়ে আর দুবার দুটি আলাদা ভালু পায় তাকে Non Repeatable Read বলে। সেমন্তে, মনে করি রাবিন একটি কেনাকাটার সাইট থেকে কোনও একটি প্ল্যাটি ৩০০ টি কিনতে চাইল, এখন তার জন্য ট্রানজেকশন A কর হল। ট্রানজেকশন A প্রথমে সেবারে যে এই পদ্ধতির ৩০০ টি স্টকে আছে কিনা। ধরে নেই সাইটের ভাটাবেজে Product_Inventory নামে একটি টেবিল আছে যাতে পদ্ধতির পরিমাণ আছে। তো ট্রানজেকশন A সেই Product_Inventory টেবিল থেকে পেল যে রাবিন যে প্ল্যাটি কিনতে চায় তা ৫০০ টি আছে। একই সময়ে জানাল এই একই সাইট থেকে ঠিক এই প্ল্যাটি ২৫০ টি কিনতে চাইল। ধরে নেই নওশিন এবং জন্য ট্রানজেকশন B কর হল এবং এই ট্রানজেকশনটি Product_Inventory টেবিল থেকে পেল যে এই পদ্ধতির ৫০০ টি স্টকে আছে। এদিকে রাবিন তার অর্ডারটিকে কনকুর্স করল তার ফলে ট্রানজেকশন A প্ল্যাটির পরিমাণ ৫০০ থাকে কমিয়ে ২০০ করে দিল এবং ট্রানজেকশন A কর্মসূচি হয়ে পেল। এদিকে জানাল তার অর্ডারে একটি পরিবর্তন করল, সে এই প্ল্যাটি বাইন দিয়ে অন্য একটি প্ল্যাটি ২৫০ টি কিনতে চাইল। ফলে ট্রানজেকশন B আবার Product_Inventory টেবিল থেকে দুজো পেল যে এই পরে অর্ডার দেয়া প্ল্যাটি মাত্র ১০০ টি রয়েছে। তাই ট্রানজেকশন B নওশিন এবং জানাল যে পরে অর্ডার দেয়া প্ল্যাটি স্টকে ২৫০ টি নেই, নওশিন আবার প্রথমে অর্ডার দেয়া প্ল্যাটি নিয়ে চাইল। এবার ট্রানজেকশন B Product_Inventory টেবিল থেকে সেবারে যে এই পদ্ধতির অবশিষ্ট আছে এবং অর্ডারটি দেয়া যাচ্ছে না। অর্থাৎ ট্রানজেকশন B একই ভাটা দুবার পড়ে দুটি ভিন্ন ভালু (পদ্ধতির প্ল্যাটি) পেল। ট্রানজেকশন B যে সমস্যাটির মুখে পড়েছে তাকে Non Repeatable Read বলে।

Phantom – মনে করি কেনাকাটার সাইটের ভাটাবেজে Order নামে একটি টেবিল আছে এবং এই টেবিলে সব অর্ডারের ভাটা আছে। এখন একজন জানতে চাইল মোট কতগুলো অর্ডার হয়েছে। তো একটি ট্রানজেকশন অ কর হল যা কিনা Order টেবিল থেকে মোট কতটি অর্ডার হয়েছে তা দেব করবে। এদিকে একই সময়ে অন্য একটি ট্রানজেকশন ই নতুন একটি ভাটা Order টেবিলে যোগ করল। এখন যদি ট্রানজেকশন অ আবার Order টেবিল থেকে মোট কতটি অর্ডার

হয়েছে তা দেব করে তাহলে তিনি ভ্যালু পাবে। এই ধরনের ঘটনাকে Phantom বলে। Non Repeatable Read এর সাথে Phantom এর পার্থক্য হচ্ছে এখনে ভাটা ভিন্ন হচ্ছে নতুন ভাটা যোগ করার ফলে কোন ভাটা মুছে ফেলার কারণে (insert or delete)।

নিচের টেবিলে কোন আইসোলেশন স্লেভলে কোন অনুবিধি জ্ঞানে হয় কী হয় না তা দেখানো হল –

Isolation Level	Dirty Read	Non Repeatable Read	Phantom
Read Uncommitted	হয়	হয়	হয়
Read Committed	হয় না	হয়	হয়
Repeatable Read	হয় না	হয় না	হয়
Serializable	হয় না	হয় না	হয় না

প্রশ্ন ২. Transaction control এর Command সমূহ লিখ।
অথবা, A transaction consists of a sequence of query and/or update statements. SQL statement must be required to end the transaction. List the SQL statements, required to end the transaction and also write their functions.***

উত্তর: Transaction control এর Command সমূহ নিম্নরূপ:

- a) Start Transaction
- b) Commit
- c) Rollback

COMMIT: COMMIT used to save the changes.
ROLLBACK: ROLLBACK used to roll back the changes.

SAVEPOINT: SAVEPOINT creates points within the groups of transactions in which to ROLLBACK.

SET TRANSACTION: It Places a name on a transaction.

Lock based protocol

প্রশ্ন ১. Lock based protocol কি? protocol সমূহের কৰ্ম দিন।**

উত্তর: **Lock based protocol** – Lock হলো একটি mechanism যা কোন data item উপর concurrent (একই সাথে সংঘটিত হওয়া) access নিরাপত্তন করে থাকে। Concurrency control মানাজার তৈরী করে। Request grant হওয়ার পরেই কেবল Transaction proceed হবে।

Data item মুছি (Q) mode & locked হতে পারে-

1. **exclusive (X) mode:** এই mode & Data item read & written উভয় হতে পারে। lock-X instruction ব্যবহার করে X-lock request করা হয়।

2. **shared (S) mode:** এই mode এ Data item শুধুমাত্র read হতে পারে। lock-S instruction ব্যবহার করে S-lock request করা হয়।

প্রশ্ন ২. Starvation কি? Starvation হওয়ার কারণ সমূহ লিখুন।

উত্তর: Starvation - Starvation হল একটা পরিস্থিতি। যখন কোন transaction কে কোন lock acquire (অর্জন/আয়ত্ত করা) করতে সমর্থন অপেক্ষা করতে হয় তখন তাকে Starvation বলে।

Starvation হওয়ার কারণ:

- locked items এর waiting scheme properly managed না হলে।
- resource leak হলে।
- একই transaction ব্যাবহার victim হিসাবে নির্বিচিত হলে।
- Concurrency control মানাজার starvation prevent পরিকল্পনা করতে পারে।

■ Two Phase Locking (2PL) Protocol: প্রোটোকলটি serializability নিশ্চিত করে। lock point এর উপর ভিত্তি করে তাদের মূলোকে serialized করার নিশ্চয়তা প্রদান করে। এটি deadlock থেকে মুক্তির নিশ্চয়তা দেয়া না।

a. **Growing (ক্রমবর্ধনান) Phase:** এই phase এ transaction lock obtain করতে পারে তবে transaction lock release করতে পারে না।

b. **Shrinking(শূরুচিত) Phase:** এই phase এ transaction lock release করতে পারে তবে transaction lock obtain করতে পারে না।

■ Strict(নির্বিচিত) two-phase : Strict two-phase protocol এ একই transaction তার সকল exclusive locks ধরে রাখে যতক্ষণ না এটি commits(সম্পর্ক)/aborts(ব্রেক) হয়। Strict Two-phase locking ব্যবহার করে Cascading roll-back avoid করা যাব।

■ Rigorous(কঠোর) two-phase locking: Rigorous two-phase locking আবারও কঠোর হয়, এখনে সমস্ত লক commits(সম্পর্ক)/aborts(ব্রেক) না হওয়া অব্যু ধরে রাখা হয়। এই protocol এ যে সমস্ত transaction commit করা হবে সে গুলো serialized order এ থাকে।

■ Timestamp Based Protocols: Timestamp based protocols এ যখন কোন transaction সিটেমে প্রবেশ করে তখন timestamp কে কেবলমাত্র একটি timestamp issue করা হয়।

W-timestamp(Q): W-timestamp(Q) হল largest time-stamp যেকোন transaction এর ক্ষেত্রে যা write(Q) অপারেশন ফলাফলে সম্পর্ক করে।

R-timestamp(Q): R-timestamp(Q) হল largest time-stamp যেকোন transaction এ ক্ষেত্রে যা read(Q) অপারেশন ফলাফলে সম্পর্ক করে।

প্রশ্ন ৩. Thomas write Rule সমূহ লিখুন।

উত্তর: **Thomas Write Rule** - Thomas Write Rule প্রোটোকলের serializability order এর প্রয়োগ করে। এটি প্রোটোকলটির প্রয়োজ্যম অর্ডারিং অ্যালগরিদমকে উন্নত করে।

Thomas Write Rule নিম্নগুলি নিম্নরূপ:

- যদি $TS(T) < Read_TS(X)$ তাহলে transaction T কর হবে এবং অপারেশন rejecte হবে সাথে rolled back ঘটবে।

- যদি $TS(T) < Write_TS(X)$ তাহলে Write_item(X) transaction সম্পর্ক হবে না কিন্তু অন্য অপারেশনের চলমান থাকে।

- যদি প্রথম দুটি এর একটিও condition না যায়, তাহলে $W_item(X)$ এই operation টি সম্পর্ক হবে এবং $W_TS(X)$ $TS(T)$ এ সেট হবে।

Deadlock Handling***

■ **Deadlock:** ভেঙে একটি নির্বিচিত পরিস্থিতি বোধ হয়ে দেখানো দুই বা ততোধিক প্রক্রিয়া একে অপরের জন্য কোন resource release এর অপেক্ষা থাকে অথবা দুই বা ততোধিক processes circular chain এ কোন resource এর জন্য অপেক্ষা করে।

এই partial schedule এর ক্ষেত্রে ধরুনঃ

T ₃	T ₄
lock-X (B)	
read(B)	
B=B-50	
read(B)	Lock-S(A)
	Read(A)
	Lock-S(B)
	Lock-X(A)

T₃ অথবা T₄কেও সংঘটিত হতে পারবে না- T₃ যতক্ষণ না B এর Lock release করবে ততক্ষণ পর্যন্ত T₄ lock-S(B) সম্পর্ক করতে পারবে না আবার T₄ যতক্ষণ না A এর Lock release করবে ততক্ষণ পর্যন্ত T₃ lock-X(A) সম্পর্ক করতে পারবে না। এই ধরনের পরিস্থিতিকে deadlock বলে।

■ **Deadlock prevention:** Deadlock prevention protocols নিশ্চয়তা প্রদান করে

■ **Deadlock Recovery:** যখন deadlock সমাপ্ত হয় তখন deadlock recovery করার দরকার হয়।

Kill process: এই পদ্ধতিতে deadlock এর সাথে জড়িত সকল process কে একে একে kill করা হয়। প্রতিটি process kill করার পর deadlock condition check করা হয়; যদকিন না system deadlock থেকে মুক্ত হয় তখন এই process repeat করা হয়।

Resource Preemption (অগ্রাধিকার): Deadlock এর সাথে জড়িত প্রক্রিয়াগুলি থেকে resources গুলো সরিয়ে ফেলা হয়, সরিয়ে ফেলা resource গুলো অন্য process এর জন্য ব্যবহার করা হয়, যাতে system deadlock থেকে মুক্ত প্রাপ্ত সম্ভাবনা থাকে।

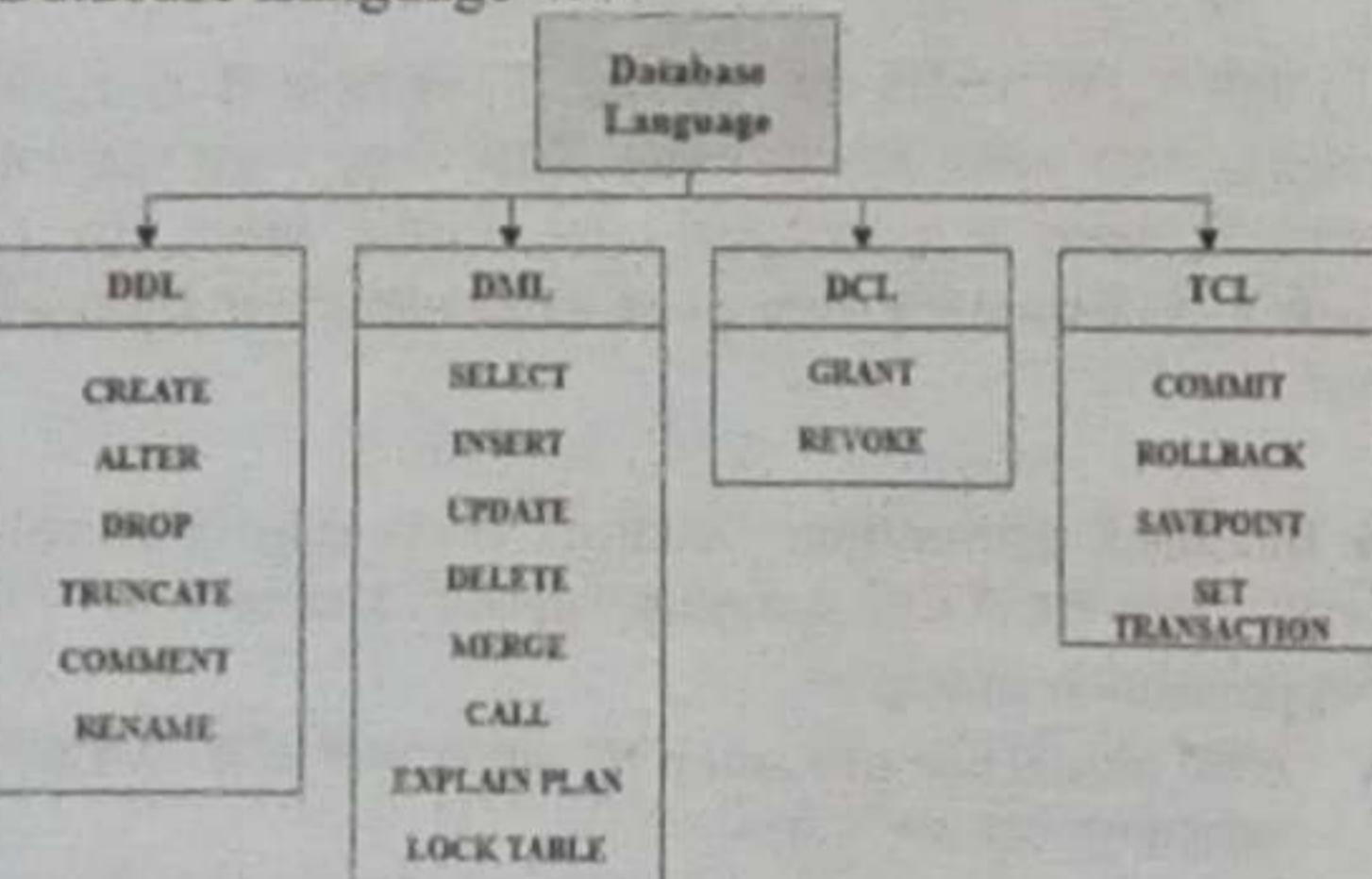
iii) Query Language (কুয়েরি ল্যাঙ্গুেজ)

প্রথ ১. Query Language কি?

উভয় Query শব্দের অর্থ তত্ত্বাংশ করা বা Search করা। যে Language এর সাহায্যে Database থেকে নির্দিষ্ট Data Insert, Delete, Modify ইত্যাদি করা যায়, তাকে Query Language বলে।

প্রথ ২. Database Language কি? এর প্রকারভেদ আলোচনা করুন। *****

উভয় Database Language - যে Language এর সাহায্যে ডাটাবেস সিস্টেম এ Schmea Definition, Query operation, Data Modification ইত্যাদি কাজ করা হয়। সে Language কে Database Language বলে।



Database Language চার ধরনের যথাঃ

- a) Data Definition Language (DDL) [BKKB-2017]
- b) Data Modification Language (DML) [BKKB-2017]
- c) Data control Language (DCL)
- d) Transaction Control Language (TCL)

■ **Data Definition Language (DDL):** যে Database Language এর মাধ্যমে Relation Database Schema কলোকে Define, Deleting, Index Creating, Modifying করা হয় এবং এর প্রয়োজনীয় command সমূহ সরবরাহ করে থাকে তাকে Data Definition Language (DDL) বলে।

DDL এর ব্যবহৃত ক্ষমতা সমূহ-

- (1) CREATE Statement
- (2) DROP Statement
- (3) ALTER Statement
- (4) RENAME Statement
- (5) Truncate Statement
- (6) Comment Statement

প্রথ ৩. DDL এবং DML এর মধ্যে পার্থক্য লিখুন।

উত্তর: DDL এবং DML এর মধ্যে পার্থক্য-

DDL	DML
১. ডাটাবেস এর মূল কাঠামো তিজাইন করার জন্য ব্যবহার করা হয়।	১. ডাটাবেস কুয়েরি অপারেশন সম্পর্ক করার জন্য ব্যবহার করা হয়।
২. এর ফলে Create, alter drop ক্ষমতা সমূহ ব্যবহৃত হয়।	২. এর ফলে Select, insert, delete, update ক্ষমতা সমূহ ব্যবহৃত হয়।
৩. এর সাহায্যে ডাটাবেস টেবিল করা হয়।	৩. ডাটাবেসের ডাটাকে ম্যানিপুলেট করা হয়।
৪. বিভিন্ন ধরনের ইন্টেগ্রেট কল্পনাইট ব্যবহৃত হয়।	৪. শর্ত সাপেক্ষে কুয়েরি করা যায়।

প্রথ ৪ . Data Definition Language এর Operation সমূহ লিখুন।

উভয় Data Definition Language এর Operation সমূহ যাহা

- a) **Create statement:** Create Statement Command দ্বারা Database table তৈরী করা হয়।

Syntax:

CREATE DATABASE databaseName
CREATE TABLE Table_name;
(Column_name datatype (Size).....)

Example:

SQL> CREATE TABLE employee (
emp_id INT NOT NULL,
name VARCHAR (25) NOT NULL,
city VARCHAR(15) NOT NULL,
PRIMARY KEY (emp_id)
);

- b) **Drop Statement:** Table মুছে ফেলার জন্য বা ডিলেট করার জন্য Drop Statement ব্যবহার করা হয়।

Syntax: DROP TABLE Table_name;

Example: employee নামের টেবিলটি ডিলেট করার জন্য নিচের ক্ষমতা ব্যবহার করি।

SQL>DROP TABLE employee

- c) **ALTER Statement:** কোন Table create করার পর পরিবর্তন, পরিমার্জন ও সংশোধন করার জন্য Alter Statement ব্যবহার করা হয়।

Syntax: ALTER TABLE table_name
ADD newcolumn name datatype (Size)

Example: employee নামক table এ join_date নামক New Column Add করার Command দেওয়া হলোঃ
SQL>ALTER TABLE employee
ADD join_date date();

- d) **RENAME Statement:** কোন Table এর নাম পরিবর্তন করার জন্য RENAME Statement ব্যবহার করা হয়।

Syntax:
RENAME old_table_name TO new_tablename;

Example: employee নামক টেবিল টিকে employee_info নামে পরিবর্তন করার জন্য নিচের ক্ষমতাঃ

SQL>RENAME employee to employee_info;

- e) **TRUNCATE Statement:** টেবিল থেকে স্কেল ডাটা মুছে ফেলার জন্য TRUNCATE Statement ব্যবহার করা হয়।

Syntax: TRUNCATE TABLE table_name;

■ **Data Manipulation Language (DML):** যে Database Language এর মাধ্যমে Relational Database এ Retrieve, Insert, Delete, Modify ইত্যাদি অপারেশন করা যায় এবং এর জন্য প্রয়োজনীয় ক্ষমতা সমূহ সরবরাহ করে থাকে, তাকে Data Maniuilation Language (DML) বলে।

DML এ ব্যবহৃত প্রয়োজনীয় ক্ষমতা সমূহ-

- a) SELECT Statement
- b) UPDATE Statement
- c) DELETE Statement
- d) INSERT Statement
- e) MERGE Statement
- f) CALL Statement
- g) EXPLAIN PLAN Statement
- h) LOCK TABLE Statement

প্রথ ৫. Data Manipulation Language (DML) কত করার ও কি কি? সংক্ষেপে লিখুন।

উত্তর (DML) দুই ধরনের যথাঃ

- a) **Procedural DML:** যে প্রকার DML এর ফলে কী ধরনের ডাটা, ডাটাবেস এ জমা হবে এবং ডাটা কীভাবে পাওয়া যাবে; তা ব্যবহারকারী কে উল্লেখ করে দিতে হয় তাকে Procedural DML বলে।

- b) **Non-Procedural DML:** যে প্রকার DML এর ফলে কী ধরনের ডাটা, ডাটাবেস এ জমা হবে তথ্মাত্ত তা ব্যবহারকারী কে উল্লেখ করে দিতে হয় তাকে Non-Procedural DML বলে।

প্রথ ৬. Procedural এবং Non-Procedural DML এর মধ্যে পার্থক্য লিখুন।

উত্তর: Procedural এবং Non-Procedural DML এর পার্থক্য-

Procedural DML **Non-Procedural DML**

1. এ ধরনের DML এর ফলে কী ধরনের ডাটা ডাটাবেস এ জমা হবে, তা ইউজার কে উল্লেখ করে দিতে হয়।
2. এ ধরনের DML এর ফলে কী ধরনের ডাটা ডাটাবেস এ জমা হবে, তা ইউজার কে উল্লেখ করে দিতে হয় না।

3. এ ধরনের DML এর ফলে কী ধরনের ডাটা প্রযোজন তা ব্যবহারকারীকে উল্লেখ করে দিতে হয়।
4. এ ধরনের DML এর ফলে কী ধরনের ডাটা প্রযোজন তা ব্যবহারকারীকে উল্লেখ করে দিতে হয় না।

5. এ ধরনের DML এর ফলে কিভাবে উক্ত ডাটা পাওয়া যাবে, তা ব্যবহারকারীকে উল্লেখ করে দিতে হয় না।
6. এ ধরনের DML এর ফলে কিভাবে উক্ত ডাটা পাওয়া যাবে, তা ব্যবহারকারীকে উল্লেখ করে দিতে হয় না।

7. এ ধরনের DML এর ফলে কিভাবে উক্ত ডাটা প্রযোজন তা ব্যবহারকারীকে উল্লেখ করে দিতে হয়।
8. এ ধরনের DML এর ফলে কিভাবে উক্ত ডাটা প্রযোজন তা ব্যবহারকারীকে উল্লেখ করে দিতে হয় না।

Data Manipulation Language এর operation সমূহ।

■ **SQL SELECT:** ডাটাবেসের একটি টেবিল থেকে ডাটা কুয়েরি করার SELECT ক্ষমতা ব্যবহার করা হয়। উদাহরণ: টেবিলের সকল ডাটা দেখার জন্য SQL ক্ষমতা-

SELECT * FROM employee_info;

emp_id	name	city	join_date
1	Kristen	London	17-12-1996
2	Russell	New York	20-1-1992
3	Albart	Los Angels	20-12-2012

■ **INSERT Statement:** Database এ Data insert /input করার জন্য INSERT Statement ব্যবহার করা হয়।

Syntax:
INSERT INTO table_name
(column_name, column_name)
VALUES (value1, value2,...)

■ **SQL UPDATE:** ডাটাবেসের একটি টেবিল থেকে কোন নির্দিষ্ট

রো কে পরিবর্তন করার জন্য UPDATE ক্ষমতা ব্যবহার করা হয়।

উদাহরণ: emp_id 3 এর সিটি পরিবর্তন এর SQL ক্ষমতা-

UPDATE employee_info SET city='New York'
WHERE emp_id=3;

SELECT * FROM employee_info;

emp_id	name	city	join_date
1	Kristen	London	17-12-1996
2	Russell	New York	20-1-1992
3	Albart	New York	20-12-2012

■ **SQL DELETE:** ডাটাবেসের একটি টেবিল থেকে কোন নির্দিষ্ট রেকর্ড DELETE করার জন্য DELETE কমান্ড ব্যবহার করা হয়।
উদাহরণ: emp_id 3 এর DELETE করার SQL কমান্ড:
DELETE FROM employee_info WHERE emp_id=3;

SELECT * FROM employee_info;

emp_id	name	city	join_date
1	Kristen	London	17-12-1996
2	Russell	New York	20-1-1992

■ **MERGE Statement:** একাধিক টেবিলের ডাটা একত্র করার জন্য MERGE Statement ব্যবহার করা হয়।

■ **SQL DISTINCT:** ডাটাবেসের একটি টেবিল থেকে নির্দিষ্ট তো এর ইউনিক value

city
London
New York

SELECT DISTINCT city
FROM employee_info;

প্রশ্ন ৭. In Which keyword is used for retrieving data from table without duplicate row data?
ডুপ্লিকেট সারি ডেটা ছাপ টেবিল থেকে ডেটা পুনরুৎসর করা কেন কীভাবে ব্যবহার করা হয়?

Answer: DISTINCT কীওয়াজটি ডুপ্লিকেট row data হচাই টেবিল থেকে ডেটা পুনরুৎসর করার জন্য ব্যবহৃত হয়।

SELECT DISTINCT Syntax

SELECT DISTINCT column1, column2, ...
FROM table_name;

■ **ORDER BY:** ডাটাবেসের কুর্সরী করা ডাটা সমূহকে কোন নির্দিষ্ট অভাবে সজানের জন্য ORDER BY ব্যবহার করা হয়। By default ascending order সজানে থাকে। উদাহরণ:
SELECT * FROM employee_info ORDER BY emp_id desc;

emp_id	name	city	join_date
3	Albert	New York	20-12-2012
2	Russel	New York	20-1-1992
1	Kristen	London	17-12-1996

ডাটাবেস থেকে নির্দিষ্ট সংখ্যক ডাটা return করার জন্য SELECT TOP clause ব্যবহার করা হয়।

[Note: SELECT TOP clause সকল database systems support করে না। MySQL support করে LIMIT clause অনুলিপি এবং করার জন্য Oracle এ ব্যবহার করা হয় ROWNUM.]

SQL Server / MS Access Syntax:

SELECT TOP number | percent column_name(s)
FROM table_name

WHERE condition;

MySQL Syntax:

SELECT column_name(s)

FROM table_name

WHERE condition

LIMIT number;

Oracle Syntax:

SELECT column_name(s)

FROM table_name

WHERE ROWNUM <= number;

■ **Group By:** Group By statement টি SQL এর কমান্ড যা ডাটাবেসের একটি টেবিল থেকে identical data কে সজানের জন্য ব্যবহার করা হয়। GROUP BY statement টি সাধারণত aggregate functions (COUNT, MAX, MIN, SUM, AVG) সমূহের সাথে ব্যবহার করা হয়।

GROUP BY Syntax:

SELECT column_name(s)

FROM table_name

WHERE condition

GROUP BY column_name(s)

ORDER BY column_name(s);

উদাহরণ:

SELECT COUNT(emp_id), city
FROM employee_info
GROUP BY city
ORDER BY COUNT(emp_id) DESC;

Count(emp_id)	city
2	New York
1	London

■ **HAVING Clause:** Aggregate functions (COUNT, MAX, MIN, SUM, AVG) সমূহের সাথে WHERE Clause ব্যবহার করা হয় না। HAVING Clause Aggregate function সমূহের সাথে Condition ব্যবহারের সুবিধা প্রদান করে।

HAVING Syntax:

SELECT column_name(s)

FROM table_name

WHERE condition

GROUP BY column_name(s)

HAVING condition

ORDER BY column_name(s);

উদাহরণ:

SELECT COUNT(emp_id), city
FROM employee_info
GROUP BY city
HAVING COUNT(emp_id) > 1;

Count(emp_id)	city
2	New York

প্রশ্ন ৮. SQL statement এ drop table এবং delete from এর মধ্যে পার্থক্য লিখুন।***

প্রশ্ন ৯. SQL statement এ drop table এবং delete from এর মধ্যে পার্থক্য:

DELETE	DROP
এটি টেবিল থেকে কিছু পরিমাণ বা সকল tuple সরিয়ে দেয়।	এটি ডাটাবেস থেকে সম্পূর্ণ ছিমা, টেবিল, ডেটামেন বা সীমাবদ্ধতাসমূহ সরিয়ে দেয়।
DELETE একটি DML এর ক্ষেত্র।	DROP একটি DDL এর ক্ষেত্র।
DELETE কমান্ডের সাথে WHERE ক্লজিটি সূলত ব্যবহৃত হয়।	DROP কমান্ডের সাথে কোন ক্লজ ব্যবহৃত হয় না।
DELETE আরা সম্পাদিত ডিজাক্সাপসমূহ বাফার ব্যবহার করে সাথে সাথে ফিরিয়ে আনা যাবে না কারণ এটি সরাসরি প্রক্রিয়া করে না।	DROP আরা সম্পাদিত ডিজাক্সাপসমূহ আবার ফিরিয়ে আনা যাবে না কারণ এটি সরাসরি প্রক্রিয়া করে না।
এটি মেমরি থেকে টেবিলের ছানকে স্মর্ণ করে না।	এটি মেমরি থেকে টেবিলের ছানকে স্মর্ণ করে।

প্রশ্ন ১০. Differ: Shared vs Exclusive Lock, Drop vs Drop Purge, Delete vs Truncate.

Ans: Difference between Shared Lock and Exclusive Lock :

Shared Lock	Exclusive Lock
Shared মোড হল একটি read-only operation	Exclusive মোড মোড লেখার পাশাপাশি রিউড অপারেশন করা হয়।
অন্যদের ডেটা আপডেট করতে পারে।	অন্যদের ডেটা পার্জ বা আপডেট করতে পার্খ দেয়।
lock-S instruction ব্যবহার করে S-lock কে অনুরোধ করা হয়।	lock-X instruction ব্যবহার করতে X-lock কে অনুরোধ করা হয়।

Difference Between Drop And Drop Purge
সম্বর্গত, একটি টেবিল রিসাইকেল দিন (as of Oracle 10g) এ সম্ভব হয়, যদি এটি dropped করা হয়। যদি purge modifier দিয়ে করা যাকে, তাহলে টেবিলটি ডাটাবেস থেকে unrecoverable dropped করা হয়।

প্রশ্ন ১০. Difference between DELETE and TRUNCATE command:***

Delete	Truncate
DELETE কমান্ডটি নির্দিষ্ট সারি (এক বা একাধিক) সূচে ফেলার জন্য ব্যবহৃত হয়।	এই কমান্ডটি টেবিল থেকে সম্পূর্ণ সারি মুছে ফেলার জন্য ব্যবহৃত হয়।
এটি একটি DML (Data Manipulation Language) ক্ষেত্র।	এটি একটি DDL (Data Definition Language) ক্ষেত্র।
আমরা DELETE কমান্ড ব্যবহার করার পরেও ডেটা রোলব্যাক করতে পারি।	আমরা রোলব্যাক করতে পারি।

DELETE কমান্ডের চেয়ে সুস্থ।

We can specify filters in where clause.

■ **Data Control Language:** DCL মানে Data Control Language। এটি stored বা saved data পুনরুত্থার (retrieve) করতে ব্যবহৃত হয়। DCL execution হল সেনদেনমূলক (transactional)। এটিতে রোলব্যাক (rollback) প্যারামিটারও রয়েছে। DCL এর সুতি ক্ষেত্র আছে।

a) **Grant:** It is used to give user access privileges to a database. [নিচের প্রশ্নে উদাহরণ সহ বিজ্ঞিপ্তি দেয়া আছে]

b) **Revoke:** It is used to take back permissions from the user. [নিচের প্রশ্নে উদাহরণ সহ বিজ্ঞিপ্তি দেয়া আছে]

প্রশ্ন ১১. Role in database & how to grant and revoke authorization user in database? ডাটাবেসের ক্ষেত্রিক এবং ডাটাবেসে অনুমোদনকারী ব্যবহারকারীকে কীভাবে অনুমোদন ও প্রত্যাহার করা যায়?***

Answer: Roles are created by users (usually administrators) to group together privileges (সুবিধা) or other roles. They are a means of facilitating the granting of multiple privileges or roles to users.

Common uses of Roles:

(i) To manage the privileges for a database application (Application Roles).

(ii) To manage the privileges for a user group (User Roles).

Application Roles: You grant an application role all privileges necessary to run a given database application. Then, you grant the secure application

role to other roles or to specific users. An application can have several different roles, with each role assigned a different set of privileges that allow for more or less data access while using the application.

User Roles: You create a user role for a group of database users with common privilege requirements. You manage user privileges by granting secure application roles and privileges to the user role and then granting the user role to appropriate users.

Grant: SQL Grant কমান্ড বিশেষভাবে ব্যবহারকারীর ব্যবহারের জন্য ভাটাবেস অবজেক্টে সুবিধা প্রদান করতে ব্যবহৃত হয়। এই কমান্ডটি ব্যবহারকারীদের অন্যান্য ব্যবহারকারীদেরও অনুমতি দেওয়ার অনুমতি দেয়।

বাক্য গঠন (Syntax):

```
GRANT privilege_name
ON object_name
TO {user_name |PUBLIC |role_name}
```

Example:

```
grant insert,
select on accounts to Candy
```

উপরের কমান্ডের মাধ্যমে Candy নামের ইউজারকে ভাটাবেস অবজেক্ট ব্যবহারের অনুমতি দিয়েছে। যেখন সে আকাউন্টে query করতে বা insert করতে পারে।

এখানে privilege_name হল কোন অনুমতি দিতে হবে, object_name হল ভাটাবেস অবজেক্টের নাম, user_name হল সেই ব্যবহারকারী যাকে আরেকে দেওয়া হচ্ছে। উচিত, পারিস্থিতিক ব্যবহার করা হয় সমস্ত ব্যবহারকারীকে আরেকে দেওয়ার জন্য।

Revoke: Revoke কমান্ড ভাটাবেস অবজেক্টে ব্যবহারকারীর বিশেষাধিকার (privileges) প্রত্যাহার করতে ব্যবহার করা হয় যদি কোন grant আগে দেয়া হয়ে থাকে। এটি grant কমান্ডের বিপরীতে কাজ করে।

Syntax:

```
REVOKE privilege_name
ON object_name
FROM {user_name |PUBLIC |role_name}
```

Example:

```
revoke insert,
select on accounts from Candy
```

উপরের কমান্ডের মাধ্যমে ব্যবহারকারী Candy এর অনুমতি দেখন query বা insert করা অনুমোদন বাদ দেয়া হচ্ছে।

প্রশ্ন 12. Example: কীভাবে একটি ভেটাবেস টেবিল থেকে অনুমতি বাতিল করা যায়? এর জন্য SQL কমান্ড লিখুন।

উত্তর: একটি নির্দিষ্ট ব্যবহারকারী অথবা সমস্ত ব্যবহারকারীদের ভাটাবেস অবজেক্টগুলিতে কার্য সম্পাদনের অনুমতি বাতিলের জন্য REVOKE স্টেটমেন্ট ব্যবহার করা হয়। কোন ব্যবহারকারীর অথবা অন্য কোন ভূমিকা থেকে কোন ভূমিকা প্রত্যাহার করার জন্য REVOKE স্টেটমেন্ট ব্যবহার করা হয়।

Revoke examples: To revoke the SELECT privilege on table t from the authorization IDs maria and harry, use the following syntax:

**REVOKE SELECT ON TABLE t FROM
maria, harry**

প্রশ্ন 13. Example: কিভাবে ভাটাবেজ এ একটি নতুন User Create করা যায়? সেই নতুন User (মনে করা যাক, RKB) কে Customer টেবিলে কিভাবে Full access দেওয়া যায়? কিভাবে সেই access ফিরিয়ে নেওয়া যায়? এ সংজ্ঞান SQL command লিখুন।

উত্তর: নিম্ন RKB নামে ইউজার ক্রিয়েট করা হয়েছে ভাটাবেজ সিন্ট্যাক্সে। সেই ইউজারকে বিটব্যাল টেবিলে সকল প্রক্রিয়া (GRANT) দেওয়া হয়েছে। শেষে সেই ইউজারকে সকল আকসেস ফিরিয়ে (REVOKE) নেওয়া হয়েছে।

```
CREATE USER RKB IDENTIFIED BY 12345
GRANT ALL ON bitBoxtable TO RKB;
REVOKE ALL ON bitBoxtable FROM RKB;
```

■ Transaction Control Language : TCL is used to run the changes made by the DML statement. TCL can be grouped into a logical transaction. Here are some tasks that come under TCL:

- Commit:** It is used to save the transaction on the database.
- Rollback:** It is used to restore the database to original since the last Commit.

VIEW (ভিত্তি): SQL এ ভিত্তি হচ্ছে এক ধরণের ভাট্যাল টেবিল। VIEW এ বাস্তব টেবিলের মত রো-কলাম থাকে। ভাটাবেসের এ টেবিল থেকে আমরা নির্দিষ্ট কিছু রো কলাম সিলেক্ট করে view টেবিল করতে পারি। view একটি টেবিলের সকল রো নিয়ে হতে পারে আবর কিছু নির্দিষ্ট কিছু রো নিয়েও হতে পারে।

VIEW এর syntax –

```
CREATE VIEW view_name AS
SELECT column1, column2.....
FROM table_name
WHERE [condition];
```

উদাহারণ: ধরি, CUSTOMERS টেবিলে নিম্নের রেকর্ড সমূহ রয়েছে। উক রেকর্ড সমূহের জন্য আমরা একটি ভিত্তি তৈরি করব।

ID	NAME	ADDRESS	SALARY
1	Jhon	Washington	15000
2	Milan	New York	21000
3	Milton	Milan	7000
4	Jack	Miami	11000

ভিত্তি তৈরি:

```
SQL > CREATE VIEW CUSTOMERS_VIEW AS
SELECT name, address
FROM CUSTOMERS;
```

ভিত্তি দেখার কমান্ড : SQL > SELECT * FROM CUSTOMERS_VIEW;

NAME	ADDRESS
Jhon	Washington
Milan	New York
Milton	Milan
Jack	Miami

প্রশ্ন 18. Define View, Materialized View, Difference btwn them and Usage of two?

Ans: view হল একটি সজিক্যাল ভাট্যাল টেবিল যা 'select' query ব্যবহার করে তৈরি করা যায়।

Materialized View গুলি মূলত সজিক্যাল ভিত্তি যা 'select' query ব্যবহার করে তৈরি করা হয়। Materialized View এর ক্ষেত্রে প্রাপ্ত ফলাফলগুলি একটি ভিস্ক বা টেবিলে সংরক্ষণ করা হয়।

View	Materialized View
View হল একটি ভাট্যাল টেবিল, এটি কোন ভাটা সংরক্ষণ করে না।	MV একটি পৃথক টেবিলে ভেটা সংরক্ষণ করে মূল টেবিলের অনুলিপি।
বিস্তারিত উদ্দেশ্যে ব্যবহার করা হয়।	কর্মসূচিতা বৃক্ষের জন্য ব্যবহৃত হয়।
বিস্তারিত মূল টেবিলটি বাদ দেওয়া হয় এবং view আরেকে করা যাবে না।	এমনকি মূল টেবিলটি বাদ দিলে MV আরেকে সংযোগ হবে।
View এ সরাসরি DML ব্যবহার করা যাবে না। কিন্তু এটি রিফ্রেশ কৌশল ব্যবহার করে আপডেট করা যেতে পারে।	MV এ সরাসরি DML অপারেশন করা যাবে না। কিন্তু এটি রিফ্রেশ কৌশল ব্যবহার করে আপডেট করা যেতে পারে।

Use of Views:

1. এটি আলিঙ্কেশনগুলিকে isolating (বিচ্ছিন্ন) করার জন্য এবং ভাটাবেস টেবিলের সংজ্ঞায় পরিবর্তন ব্যক্ত করার জন্য ব্যবহার করা যেতে পারে।
2. View কে SQL statement সরলীকরণের জন্য ব্যবহার করা যেতে পারে আরও ভাল এবং সুত ব্যবহারের জন্য।

3. row এবং column এর পূর্বনির্ধারিত সেটগুলিতে restricted (সীমাবদ্ধ) আরেকের মধ্যে নিরাপত্তা বাস্তবায় জন্যও View ব্যবহার করা হয়।

Use of Materialized Views:

1. Materialized view বিভিন্ন querry এর কর্মসূচিতা ভৱিত করতে ব্যবহার করা যেতে পারে, যার মধ্যে রয়েছে ভেটা একটা ক্রাক্স(aggregations) এবং রূপান্বয়(transformation)।

2. updatable materialized view এর সাহায্যে ভেটা সরিবেশ করা, মুছে ফেলা, আপডেট করা এবং ঘোষ করা সম্ভব(insert, delete, update and add data ইত্যাদি)।

প্রশ্ন 15. Query Language এর প্রকারভেদ উল্লেখ করুন।

উত্তর: Query Language কে প্রধানত দুই ভাগে ভাগ করা যায়।

- a) Procedural Query Language
- b) Non-Procedural Query Language
- c) Formal Query Language
- d) Commercial Query Language

প্রশ্ন 16. Procedural & Non-Procedural Query Language কি?

উত্তর: Procedural Query Language: ভাটাবেস এর ধারাবাহিক অপারেশন সম্পর্ক করার জন্য যে Language ব্যবহার করা হয় তাকে Procedural Query Language বলে।

Non-Procedural Query Language: Database এর কাজিক্ত Information কে বর্ণন করার জন্য যে Language ব্যবহার করা হয় তাকে Non-Procedural Query Language বলে।

প্রশ্ন 17. Query Language এর কাজ লিখুন।

উত্তর: Query Language এর কাজ হলো:

- a) Database থেকে নির্দিষ্ট ভাটা কে খুঁজে বের করা।
- b) ভাটা কে retrieve করা।
- c) ভাটা কে মডিফাই করা।
- d) ভাটা কে organize করা ইত্যাদি।

প্রশ্ন 18: Formal & Commercial Query Language কি?

উত্তর: Formal Query Language: যে সম্ভ �Query Language এ Procedural & Non-Procedural Query Language এর মধ্যে কেন একটি সুবিধা প্রদান করে, তাকে Formal Query Language বলে।

Commercial Query Language: যে সম্ভ Query Language এ Procedural & Non-Procedural Query Language এর সুবিধা প্রদান করে, তাকে Commercial Query Language বলে।

প্রশ্ন 19. কয়েকটি ফর্মাল Query Language এর নাম লিখুন।
উত্তর: Formal Query Language এর নাম হলোঃ

- ক) The relation Algebra
- খ) The tuple Relational calculus
- গ) The Domain Relational Calculus ইত্যাদি।

প্রশ্ন 20. কয়েকটি Commercial Query Language এর নাম লিখুন।
উত্তর: Commercial Query Language নাম হলো হলোঃ

- ক) 4732 Query
- খ) Access Query Language
- গ) Chess Query Language
- ঘ) Contextual Query Language
- ঙ) Data log
- চ) IBM Query Management Facility

প্রশ্ন 21. Commercial Query Language এর সুবিধা হলো লিখুন।

উত্তর: Commercial Query Language এর সুবিধা হলোঃ
ক) ইহা User Friendly।
খ) ইহা ব্যবহার করে সহজ ও সক্ষতার সাথে ডাটাবেস কূরোরি অপারেশন করা যায়।
গ) ইহা Procedural ও Non-Procedural
Language এর সুবিধা প্রদান করে।
ঘ) ইহা ডাটাবেস এ ডাটা কে ডিফাইন, মডিফাই ইত্যাদি কাজ করে থাকে।
ঙ) ইহাতে ডাটার নিরাপত্তা সরচেরে বেশি।

প্রশ্ন 22. কয়েকটি জনপ্রিয় কূয়েরি Language এর নাম লিখুন। [36 BCS]**

উত্তর: কয়েকটি জনপ্রিয় কূয়েরি Language এর নাম নিম্নরূপঃ
ক) SQL
খ) My SQL
গ) QBE (Query by Example)
ঘ) QUEL (Query Language)
ঙ) PostgreSQL ইত্যাদি।

প্রশ্ন 23. SQL কি? [36 BCS]

উত্তর: SQL: Structural Query Language কে সংক্ষেপে SQL
বলে। ইহা একটি শক্তিশালী Data Definition ও Manipulation
Language।

প্রশ্ন 24. SQL এর বৈশিষ্ট্য লিখুন।

উত্তর: SQL এর বৈশিষ্ট্য হলোঃ
ক) SQL একটি Text based Languaged।
খ) এতে SELECT, INSERT, DELETE শব্দসমূহ দ্বাৰা Command Set তৈরি করা হয়েছে।

প্রশ্ন 25. SQL এর Basic Domain type কি? Basic Domain কি? Basic Domain এর নাম লিখুন।
উত্তর: SQL বিভিন্ন ধরনের ডাটা টাইপ সার্পোর্ট করে এগুলোকে SQL Basic Domain বলে। মূলত Basic Domain SQL এ Built in data type.

- ক) Clear (n)
- খ) Varchar (n)
- গ) int (n)
- ঘ) Numeric (p,d)
- ঙ) প্রিসিশানসহ Fixed point প্রকাশ করে।
Example Number (3.1) p = 3, d = 1
99.9 পর্যন্ত প্রকাশ করা যাবে কিন্তু 444.4 লিখা যাবে না।

প্রশ্ন 26. Database Manager এর অংশগুলো কি কি?

উত্তর: Database Manager এর অংশগুলো হলোঃ
ক) Transection Manager
খ) File Manager
গ) Buffer Manager
ঘ) Authorization Manager

প্রশ্ন 27. SQL এর বিভিন্ন অংশের নাম লিখুন।

উত্তর: SQL এর বিভিন্ন অংশের নাম হলোঃ
ক) DML: (পূর্বের অংশে উল্লেখিত)
খ) Integrity: Integrity constants সুনির্দিষ্ট করার জন্য ব্যবহার করা হয়।
গ) View definition: View design এর জন্য প্রয়োজনীয় Command সরবরাহ করে।
ঘ) Transaction control
ঙ) Embedded SQL
ঘ) Dynamic SQL
ঘ) Authorization

প্রশ্ন 28. Query Processor কি?

উত্তর: Query Processor মূলত Query Lanugage এ দিইত Statement সমূহকে Low-Level instruction এ অনুবাদ করে দায়িত্ব পালন করে থাকে।

প্রশ্ন 29: QBE & QUEL কি?

QBE (Query By Example): এই শ্যাখায়ে টি QBE কোমেইন ক্যালকুলেসের উপর ভিত্তি করে তৈরি করা হয়েছে এবং এ বিমাতিক সিন্টেক্স আছে।

QUEL (Query language): এই শ্যাখায়ে সেট অপারেশন যোগন: ইউনিয়ন, মাইনাস, ইন্টারসেকশন ইত্যাদি অপারেশনের মাধ্যমে কূয়েরী সম্পাদিত হয়।

প্রশ্ন 30. SQL এর Set ক্ষমতা অপারেশনগুলো কি কি?

- উত্তর: SQL এর Set ক্ষমতা অপারেশনগুলো হলোঃ
ক) Union minus operation
খ) Intersect operation
গ) Except operation
ঘ) Union all operation

প্রশ্ন 31. SQL Sentence গঠনের নিয়ম লিখুন।

উত্তর: SQL Sentence গঠনের নিয়ম নিম্নরূপঃ
ক) SQL Command এক বা একাধিক Line এ লেখা যায়।
খ) একটি SQL Statement তর করতে হয় Verb দিয়ে। এ Verb
এর সাথে যুক্ত থাকে nouns এবং Adjectives।
গ) অভ্যোক্তি Verb এর অধীনে এক বা একাধিক Clauses থাকে।
ঘ) অভ্যোক্তি Clauses এর অধীনে এক বা একাধিক Parameters থাকে।
ঙ) Clauses এর মধ্যে Parameters কে পৃথক করতে Comma দিতে হয়।
ঘ) SQL-Statement দোষ করাতে; (সেমিকোলন) ব্যবহার করতে হয়।
ঘ) SQL-Statement upper case বা lower case এ লেখা যায়।

প্রশ্ন 32. Data Integrity কত প্রকার ও কি কি?

উত্তর: Data Integrity তিনি প্রকার। যথাঃ
ক) Entity Integrity
খ) Domain Integrity
গ) Referential Integrity

প্রশ্ন 33. SQL expression এবং Clause কয়টি ও কি কি? বর্ণনা করুন।

উত্তর: SQL expression এবং Clause তিনটি। যথাঃ
ক) Select Clause: Select Clause এর মাধ্যমে টেবিল এর এক
বা একাধিক কলাম বা রো কে সিলেক্ট করা যায়।

Syntax: SELECT*

or,
SELECT Column 1, Column 2,.....n

এখন, *(astarisk) চিহ্নটি টেবিল এর সকল কলাম এবং রো কে
সিলেক্ট করে। আর নির্দিষ্ট কোন কলাম কে সিলেক্ট করতে চাইলে
SELECT লিখে Column এর নাম লিখতে হয়।

ঘ) From: এর মাধ্যমে Table নির্বাচন করা হয়।

Syntax:- FROM table_name;

ঙ) Where:- Condition Apply করা হয়।

Syntax:- WHERE column=value;

প্রশ্ন 34. WHERE Clause এ কোন ভাবে ডাটা নিয়ন্ত্রণ করা যায়।

উত্তর: WHERE Clause এ দুই ভাবে ডাটা নিয়ন্ত্রণ করা যায়। যথাঃ
ক) কোন বিশেষ শ্রেণীর ডাটা SELECT করে।
খ) একাধিক টেবিল থেকে ডাটা সিলেক্ট করে।

প্রশ্ন 35. Formal ও Commercial Query Lanuage এর মধ্যে পার্থক্য লিখুন।

উত্তর: Formal ও Commercial Query Lanuage এর মধ্যে
পার্থক্য নিম্নরূপ।

Formal Query Lanuage	Commercial Query Lanuage
১. যে Query Lanuage এ Procedural অথবা Non-Procedural সুবিধা প্রদান করে, তাকে Formal Query	১. যে Query Lanuage এ Procedural অথবা Non-Procedural উভয় সুবিধা প্রদান করে, তাকে

Lanuage বলে। (যে কোন একটি)	Commercial Lanuage বলে।
২. ইহা User Friendly নয়।	২. ইহা User Friendly।
৩. একে Commercial কাজে Use করা যায় না।	৩. একে Commercial কাজে Use করা যায়।
৪. এটি শিক্ষা ক্ষেত্রে ব্যবহার করা কঠিন।	৪. ইহা শিক্ষা ক্ষেত্রে ব্যবহার করা সহজ।
৫. বিলেশনাল অ্যালজেব্রা ও টুপল বিলেশনাল Calculas হল এর উদাহরণ।	৫. SQL, QUEL, QBE হল এর উদাহরণ।

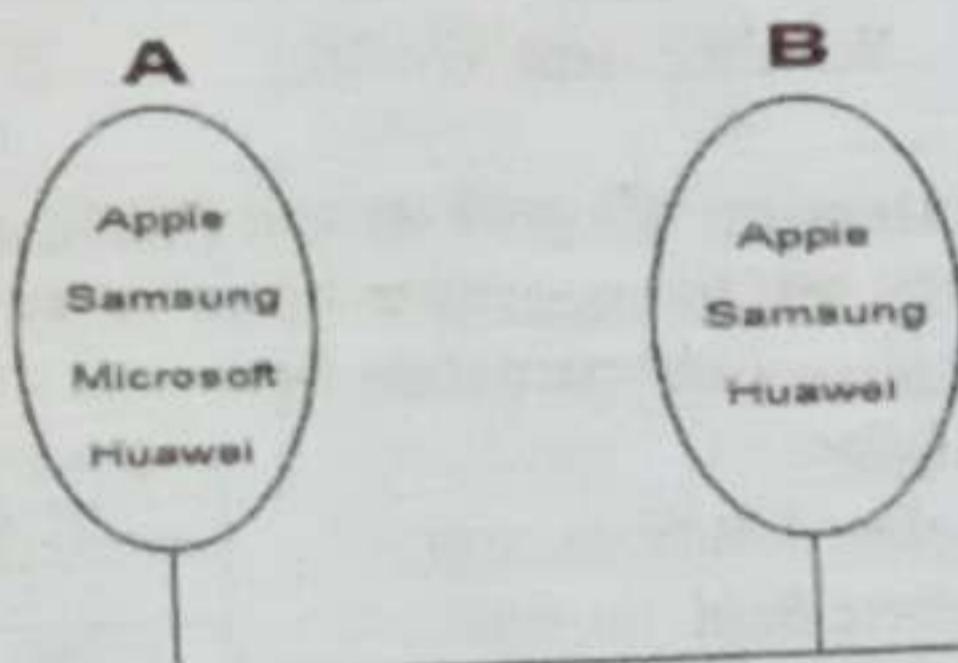
প্রশ্ন 36. SQL Set operation কি?

উত্তর: SQL Set operation: দুই বা ততোধিক Select statement হতে একত্রিতভাবে ফলাফল উপস্থাপন করার জন্য SQL Operation ব্যবহার করা হয়। SQL Operation তিনি হলোঃ
ক) UNION ALL যাকে "U" দ্বাৰা চিহ্নিত কৰা হয়।
ঘ) UNION Small "u" দ্বাৰা চিহ্নিত কৰা হয়।
গ) Intersect "∩" দ্বাৰা চিহ্নিত কৰা হয়।
ঘ) EXCEPT/MINUS "-" দ্বাৰা চিহ্নিত কৰা হয়।

UNION ALL,
UNION,
INTERSECT

এবং MINUS

কোজ এর মধ্যে
সম্পর্ক চিত্ৰে
মাধ্যমে
সহজে
উদাহৰণে
কৰা
হল।



Union all: Apple, Samsung, Microsoft, Huawei
Apple, Samsung, Huawei

Union: Apple, Samsung, Microsoft, Huawei

Intersect: Apple, Samsung, Huawei

Minus: Microsoft

Emp Table:

Job	Dept. no
Manager	10
Salesman	30
Clerk	10
Salesman	30
President	10
Salesman	30
Manager	30
Clerk	30

Union all: Emp table 10 no dept এবং 30 no dept এবং job Union all এর মাধ্যমে সেখানে হল।

SQL>
SELECT job
FROM emp
WHERE dept_no = 10
UNION ALL
SELECT job
FROM emp
WHEN dept_no = 30;

Job
Manager
Clerk
President
Salesman
Salesman
Salesman
Manager
Clerk

Union: দুটি Query এর মধ্যে Duplicate বাকি সকল Value নির্বাচন করে। পৃষ্ঠা Query Union এর মধ্যে লেখানো হলো।

SQL>
SELECT Job
FROM emp
WHERE dept_no = 10
UNION
SELECT Job
FROM emp
WHERE dept_no = 30;

Job
Clerk
Manager
President
Salesman

Intersect: দুটি কূড়োর এর মধ্যে Common উপজেলার নির্বাচন করার জন্য Intersect কোজ ব্যবহার করা হয়। পৃষ্ঠা উদ্ধৃতিত কূড়োর Intersect এর মধ্যে লেখানো হলো।

SQL>
Select Job from emp
where dept_no = 10
INTERSECT
Select Job From emp
Where dept_no = 30

Job
Clerk
Manager

Expect/Minus: দুটি কূড়োর এর মধ্যে অবশ্যই কূড়োর থেকে ২য় কূড়োর সমান্তর বাকি নিয়ে ব্যক্তিগতে অন্তর্ভুক্ত জন্য Minus কোজ ব্যবহার করা হয়।

উদাহরণ
SQL>Select Job from emp
where dept_no = 10
MINUS
Select Job from emp
Where dept_no = 30

Job
President

এগ ০৯. In a SQL query, while performing string matching when do we use operator and when we use LIKE operator? Give examples.

এসকিউএল কোয়ার্ইটে, স্ট্রিং বাচিং করার সময় আমরা কখন অপারেটর ব্যবহার করব এবং কখন LIKE অপারেটর ব্যবহার করব? উদাহরণ নাও।

উত্তর: String matching operator: শর্ত দেখায় থাকলে নির্দিষ্ট একটি স্ট্রিং করার জন্য String matching operator ব্যবহার করা হয়। উদাহরণসমূহ: যদি আমরা Noakhali নামের একটি নির্দিষ্ট উপজেলা থেকে কস্টমার সূজে বের করতে চাই। তাহলে এইভেজে আমরা String matching Operator ব্যবহার করতে পারি।

C_id	Name	Upzilla	District	Post_code
1	Anisha	Kurigram	Kurigram	3000
2	Harun	Kurigram	Kurigram	1000
3	Saifulla	Noakhali	Noakhali	3500
4	Kibria	Tangail	Tangail	3300

SELECT * FROM Customers
WHERE Upzilla = 'Noakhali';

Output

C_id	Name	Upzilla	District	Post_code
3	Saifulla	Noakhali	Noakhali	3500

LIKE Operator: SQL এর মধ্যে standard pattern matching পদ্ধতি করে LIKE operator যা সকলসময় WHERE clause এর পরে ব্যবহার করা হয়। wildcard characters ব্যবহার করে এন্ডেন্ড কিছু কভিশনের উপর ভিত্তি করে ইয়ে যেকোন pattern matching করতে পারে। LIKE operator শৈল বা অধিক কার্যকোষ একাশ করে থাকে। যেমন:

SELECT * FROM Customers
WHERE Upzilla LIKE '%';

Output

C_id	Name	Upzilla	District	Post_code
3	Saifulla	Noakhali	Noakhali	3500
4	Kibria	Norail	Norail	3300

এগ ১৮. What is SQL Constraint? Why uses constraint? Differ table and column level Constraint? [এসকিউএল সীমাবদ্ধতা কি? কেন সীমাবদ্ধতা ব্যবহার করা হয়? টেবিল এবং কলাম ত্বরের সীমাবদ্ধতার মধ্যে পার্থক্য কি?]

উত্তর: SQL constraint শুলি একটি টেবিলে ভোর জন্য নির্দিষ্ট করতে ব্যবহৃত হয়। একটি টেবিলে সূত হতে পারে এমন ভোর type limit করতে Constraint ব্যবহার করা হয়। এটি টেবিলের ভোর নির্ভুলতা এবং নির্ভরযোগ্যতা নিশ্চিত করে। যদি constraint এবং data action এর মধ্যে কোনো violation থাকে, তখন সেই আক্ষেপ বাতিল করা হয়।

Table level Constraint (সীমাবদ্ধতা) এবং column level Constraint (সীমাবদ্ধতা) এর মধ্যে পার্থক্য: Table Constraint এবং column Constraint একই কাজ; পার্থক্য হল কেবলের অধিন জৰুর উত্তের ক্ষেত্ৰে। Table constraint আপনাকে একটি PRIMARY KEY, UNIQUE, CHECK, অথবা FOREIGN KEY এর মধ্যে একাধিক কলাম সpecify করার অনুমতি দেয়। Column-level constraint (CHECK constraint স্বীকৃত) শুধুমাত্র একটি কলাম refer করে।

SQL Sub Query

■ SQL Sub Query: যখন একটি কূড়োর মধ্যে আরেকটি কূড়োর Where clause এর মধ্যে Embaded অবস্থার থাকাকে Sub Query বলে।

Syntax:

SELECT column_name
FROM table_name
WHERE column_name
expression_operator

>Main Query

(
SELECT column_name
FROM table_name
WHERE condition
);

Sub Query

Execution এর মধ্যে Sub-Query এবং পরে Main Query execute হবে।

(SELECT max(salary) FROM employees
WHERE City='Dhaka');

ID	Name	Salary
5	Kathrin	8500

উদাহরণ ৫: employees টেবিল থেকে ঐ সময় employee দেখার SQL command শিখন যাদের salary Dhaka এর employee দের গড় salary থেকে বেশি।

SELECT * FROM employees

WHERE salary >

(SELECT AVG(salary) FROM employees);

ID	Name	Age	City	Salary
5	Kathrin	34	Bangkok	8500
7	Jackson	25	Dhaka	7000

উদাহরণ ৬: Customers এবং Orders নামক ২টি রিলেশন দেখায় আছে। রিলেশনগুলো থেকে নির্দেশ অপারেশনগুলো সম্পর্ক কূলুন Customers

CID	Name	City	Country
1	Maria	Berlin	Germany
2	Aria Cruz	São Paulo	Brazil
3	Victoria	London	UK
4	Howard Snyder	Eugene	USA
5	Carine Schmitt	Nantes	France
6	Elizabeth Brown	London	UK
7	Hanna Moos	Mannheim	Germany
8	Yoshi Latimer	Elgin	USA
9	Rene Phillips	Portland	USA

OID	CID	EmployeeID	ShipperID
1	3	31	101
2	2	3	4
3	1	12	4
4	6	24	42
5	8	10	1
6	4	7	4
7	5	9	5
8	7	101	10
9	9	16	10
10	8	10	4

উদাহরণ ৭: USA থেকে আজোর্ক অর্ডার সকল Customers দেখার SQL command শিখন।

SELECT * FROM customers

WHERE CID IN (

SELECT CID

FROM orders

WHERE Customers.Country = 'USA'

);

CID	Name	City	Country
4	Howard Snyder	Eugene	USA
8	Yoshi Latimer	Elgin	USA
9	Rene Phillips	Portland	USA

উদাহরণ ৬: এ সমস্যা Customers দের মধ্যে SQL command শিখুন যাদের একাধিক অর্ডার রয়েছে।

```
SELECT * FROM Customers
WHERE CID IN (
    SELECT COUNT(CID) FROM orders
    GROUP BY CID
    HAVING COUNT(CID)>1);
```

CID	Name	City	Country
8	Yoshi Latimer	Elgin	USA

উদাহরণ ৭: Programming এর বিভিন্ন কোর্স এবং কোর্সের প্রশিক্ষক এর তথ্য ডাটাবেজে এ নির্দল টেবিল সৃষ্টি সংরক্ষণ করে, course(course_id, course_name, chief_id), instructor(instructor_id, course_id), salary(joining_date)

বিভিন্ন কোর্সের প্রশিক্ষকের গত স্থানের মধ্যে SQL লিখুন প্রতিটি কোর্সের জন্য কোর্সের নাম এবং প্রশিক্ষকের গত স্থান কৈবল্য।

উত্তর

```
Select avg(i.salary), c.course_name,
From course c innerjoin instructor I ON
c.course_id = i.course_id
Group by c.course_id
```

৩ Relational Algebra

প্রশ্ন ১. Relational Algebra কলতে কি বুঝাই?*

উত্তর: Relational Algebra: Relation Algebra হল একটি Procedural কুয়েরি Language। এটি একসেট operation নিয়ে গঠিত যা এক বা একাধিক রিলেশন ইন্সুট হিসেবে প্রয়োজন করে এবং একটি নতুন রিলেশন কে আউটপুট হিসেবে প্রেরণ করে।

প্রশ্ন ২. Relational Algebra এর মৌলিক / বেসিক / Fundamental operation তালোর নাম লিখুন।

উত্তর: Relation Algebra এর মৌলিক operation তালো হল:

- a) Select
 - b) Project
 - c) Union
 - d) Set Difference
 - e) Cartesian Product
 - f) Rename
- [এতোকে একটি রিলেশন বলে]

■ Extra Relational Operation

- a) Set intersection
- b) Natural Join
- c) Division
- d) Assignment

প্রশ্ন ৩. ডাটাবেস আপডেটিং বলতে কি বুঝাই?

উত্তর: কোন ডাটাবেস এ ডাটা মুছা, নতুন ডাটা সংযোজন এবং ডাটা এর আংশিক পরিবর্তন করার প্রক্রিয়াকে ডাটাবেস আপডেটিং বলে।

- প্রশ্ন ৪. Relational Database/ Data model এর সুবিধা লিখুন।
- উত্তর: Relational Database এর সুবিধা সমূহ ক। এতে ডাটা সমূহ টেবিল আকারে থাকে।
- খ) কুয়েরি করা সহজ।
 - গ) Insert, Delete & Modification করা সহজ।
 - ঘ) এতে Relation Algebra ব্যবহার করা হয়।
 - ঙ) সময় কম লাগে।

প্রশ্ন ৫. Relational Algebra এর ব্যবহার লিখুন।

- উত্তর: Relational এলজেব্রা এর ব্যবহার তালো হলোঃ
- ক) Procedural Query Language এর জন্য Relational Algebra ব্যবহৃত হয়।
 - খ) Operation এর জন্য এক বা একাধিক রিলেশন কে L/P হিসেবে ব্যবহৃত হয় এবং একটি রিলেশন কে O/P হিসেবে প্রদান করা হয়।

প্রশ্ন ৬. Unary operation কত প্রকার ও কি কি বর্ণনা করুন।

উত্তর: Unary operation ২ প্রকার যথা-

- a) Select operation: কোন একটি রিলেশনের inverted comma (" ") এর ভিতরের প্রস্তুত শর্তকে সমর্থন করে এমন একটি tuple সিলেক্ট করার ক্ষেত্রে সিলেক্ট অপারেশন ব্যবহৃত হয়। একে O (সিঙ্গেল) দ্বারা প্রকাশ করা হয়।

Syntax:

$\sigma \text{ predicate_name} = "condition" (\text{table-name})$

↑
Predicate Part

information			
Name	Dept	ID	C GPA
Arif	ME	194044	3.80
G.Kibria	CSE	3134061	3.25
Harun	CSE	134011	3.57
Santa	CE	184009	3.86
Swapon	CSE	4134020	3.75
Saiful	CSE	214027	3.45

Ex 1. যাদের Dept, CSE তার জন্য সিলেকশন অপারেশন দেখাও।

উত্তর: $\sigma \text{ Dept} = "CSE" (\text{information})$

Name	SID	CGPA
G.Kibria	3134061	3.25
Harun	134011	3.57
Swapon	4134020	3.75
Saiful	214027	3.45

[Predicate এ $=, \neq, \leq, \geq, <, >$ ব্যবহার করা যাবে তাহাতে সুই বা তঙ্গের প্রয়োজনীয় এক্রিপ্ট যাচাই করে প্রদর্শন করার জন্য প্রয়োজনীয় অপারেশন ব্যবহৃত হয়। একে (π) পাই দ্বারা প্রকাশ করা হয়।]

- b) Project operation: কোন Relation এর একাধিক এক্রিপ্ট হতে প্রয়োজনীয় এক্রিপ্ট যাচাই করে প্রদর্শন করার জন্য প্রয়োজনীয় অপারেশন ব্যবহৃত হয়। একে (π) পাই দ্বারা প্রকাশ করা হয়।

Syntax: $\pi \text{ Attribute}_1, \text{Attribute}_2, \dots, \text{Attribute}_n (\text{table name})$

Ex 2. পূর্বের টেবিল থেকে Name এবং Dept দেখায় প্রয়োজন

অপারেশন (π) দেখাও।

উত্তর: π Name, Dept (information)

Name	Dept
Arif	ME
G.Kibria	CSE
Harun	CSE
Santa	CE
Swapon	CSE
Saiful	CSE

Rename Operation: Rename operation এর মাধ্যমে

Algebra এক্রেশনে নাম দেওয়া হয়। একে ρ_x দ্বারা প্রকাশ করা হয়।

Syntax: ρ_x(E)

এখানে, E -এক্রেশন এবং x এক্রেশন যার মাধ্যমে E এর ফলাফল পাওয়া যাবে।

প্রশ্ন ৭. রিলেশনাল ডাটাবেস আপডেটিং এর ফাংশন সমূহ কি? এবং তাদের সংক্ষিপ্ত বর্ণনা দাও।

উত্তর: রিলেশনাল ডাটাবেস আপডেটিং এর ফাংশন সমূহ হলোঃ

ক) ডিলেশন (Deletion) খ) ইনসার্চন (Insertion)

গ) অপডেটিং/মডিফিকেশন (Updating or Modification)

রিলেশনাল ডাটাবেস আপডেটিং এর ফাংশন সমূহের বর্ণনা নিম্নরূপঃ

ক) ডিলেশন (Deletion): Relational Database এ Delete Request SQL কুয়েরির মত। তবে এতে tuple প্রদর্শন না করে তথ্যমাত্র সিলেক্টেড tuple সমূহকে ডাটাবেস হতে মুছে ফেলা হয়।

Syntax: - r ← r — E

এখানে, r-relation

"←" আসাইমেট অপারেশন

"—" সেট ডিফারেন্স

Ex 1. "Rose" এর সমস্যা ডাটা ডিলেট করায় expression.

Data ← Data — σ Name = "Rose" (info)

(এখানে Data — Relation name)

Ex 2.: মনে করি একটি table আছে এর মধ্যে

আকাউন্ট এ 50tk এর নিচে/সমান আছে এ সমস্যা আকাউন্ট ডিলেট

করার expression লিখ।

Expression: loan ← loan — σ account ≤ 50 (loan)

৩) Insertion: Relational Database এর কোন relation এ

Data insert করার জন্য এই কুয়েরি রচনা করা হয়। একের প্রয়োজন হলোঃ Insertion Tuple Attribute Relation Domain এর সদস্য হতে হবে এবং একই Value হতে হবে।

Syntax: r ← r ∪ E (এখানে ∪ Union operation)

Ex 1. মনে করি Tamim নামক একজন কাস্টমার এর GEC

HB-100 Account এ 100000 টাকা জমা রাখতে হবে।

Account ← Account ∪ {"GEC", HB-100, 100000}

deposit ← deposit ∪ {"Tamim", HB-100}

Ex 2. Consider the Electrical Power company database which has the following tables: {BPDP-2021-BUET}

Powerplant(Powerplant_ID, location,

type, capacity, unit_price)

Customer(Customer_ID, name,

address, DoB, monthly_demand)

Customer_usage_profile(ID,

month_name, Customer_ID, Powerplant_ID)

- The powerplant relation has attributes powerplant_ID, location, Type{Thermal power, hydro power, nuclear power, capacity, and unit_price of power generated by the powerplant.}
- The customer relation has attributes Customer_ID, name, address, date of birth (DoB) and monthly_demand of electrical power.
- The customer_usage_profile relation stores the user profile of a customer. A customer more usage hydropower during the rainy season and thermal or nuclear power during the dry season.

প্রশ্ন ৮. Write the relational algebra expressions for the following

প্রশ্ন ১০. Aggregate Function বর্ণনা !***
উভয় Aggregate Function বর্ণনা নিম্নরূপ:-**

Function	কাজ
Sum	একসেট রাশির সমষ্টি নির্ণয় করে।
Avg	একসেট রাশির গড় নির্ণয় করে।
Max	একসেট রাশির মধ্যে বৃহত্তম রাশি নির্ণয় করে।
Min	একসেট রাশি মধ্যে ক্ষুদ্রতম রাশি নির্ণয় করে।
Count	রাশির সংখ্যা গণনা করে।

Sum: একসেট Value এর সমষ্টি নির্ণয় করে Sum Function ইনপুট হিসেবে Numerical Value প্রদান করে এবং Numerical Value Output প্রদান করে।

উদাহরণটা কোন ব্যাংকের আকাউন্ট রিলেশন এর বলশান শাখায় সকল ব্যালেন্স এর যোগফল নির্ণয় করার জন্য Sum Function ব্যবহার করে।

SQL এক্সেন্সনটি নিম্নরূপ:

```
SELECT sum(balance)
FROM account
WHERE brunch_name = "Gulshan"
```

Avg: এর মাধ্যমে Relation Attribute এর একাধিক Value এর গড় নির্ণয় করে। উদাহরণটা Account relation হতে "গুলশান" শাখায় balance এর গড় নির্ণয় করার জন্য Avg Function টি নিম্নরূপ।

```
SELECT Avg(balance) FROM account
WHERE brunch_name= "Gulshan";
```

Max: একসেট Value এর মধ্যে Maximum Value মান প্রদান করে থাকে। এটি Numeric & Non-numeric উভয় Value উপর কাজ করে।

উদাহরণটা HSBC ব্যাংকের account relation বলশান শাখায় সর্বোচ্চ ব্যালেন্স নির্ণয় করার Max Function টি নিম্নরূপ।

```
SELECT MAX(balance) FROM account;
WHERE brunch_name= "Gulshan";
```

Min: এ ফাংশনটি একসেট অপারেশন এর মধ্যে মিনিমাম ভ্যালু প্রদান করে থাকে।

উদাহরণটা DBBL ব্যাংকের Account Relation ধানমতি শাখায় সর্বনিম্ন ব্যালেন্স নির্ণয় করার Min এক্সেন্সনটি নিম্নরূপ।

```
SELECT Min(balance)
FROM account;
```

Count: ফাংশনটি একটি টেবিল থেকে নির্দিষ্ট শর্ত সাপেক্ষে মোট row এর সংখ্যা return করে।

উদাহরণটা DBBL ব্যাংকের Account Relation ধানমতি শাখায় যাদের balance ১০০০০০ উপরে তাদের মোট সংখ্যা নির্ণয় করার count এক্সেন্সনটি নিম্নরূপ।

```
SELECT COUNT (balance)
FROM account
WHERE balance > =100000;
```

PL SQL

প্রশ্ন ১. PL/SQL সর্বকালে বিশেষ বর্ণনা দিন।**

উত্তর: PL/SQL - PL/SQL এর পূর্ণরূপ হচ্ছে Procedural Language for Structured Query Language। ইহা SQL এর সাথে Programming Language এর Procedural feature এর মিলিত সংস্করণ। ইহা SQL এবং Oracle Relational Database এর জন্য Oracle Corporation এর Procedural extension যেমন- Oracle Database (version 6 থেকে), Times Ten in-memory database (version 11.2.1 থেকে), IBM DB (version 9.7 থেকে) তে PL/SQL বিদ্যমান।

Advantage: It can execute more than one or two DML and other instruction.

PL/SQL এর structure:

```
Declare
[declaration of variables]
Begin
[Main instruction]
Exception
[Handle any error message]
End; [End of Program]
```

Section গুলোর বর্ণনা নিম্নরূপ:

Declaration section: এই section টি DECLARE keyword দিয়ে আরম্ভ হয়। এটি একটি অপশনাল সেকশন। Program এ ব্যবহৃত সব ধরনের variable, cursor, subprogram এবং অন্যান্য উপাদান এই section এ define করা হয়।

Executable Section: এই section টি BEGIN keyword দিয়ে শুরু এবং END keyword দিয়ে শেষ হয়। এই section টি PL/SQL program এর mandatory। এই section এ কমপক্ষে একটি কার্য নির্বাচন (executable) line থাকবে।

Exception Handling Section: এই section টি EXCEPTION keyword দিয়ে শুরু হয়। ইহাও একটি অপশনাল সেকশন, যা program এর error নিয়ন্ত্রণ করে।

উদাহরণ ১: To print any message or value in PL/SQL

PL/SQL কোড:

SQL> Set serveroutput On

SQL> Declare

```
a number(10);
b number(10);
c number(10);
```

Begin

```
a := 10;
b := 20;
c := a+b;
```

```
dbms_output.put_line('Total');
dbms_output.put_line(c);
End;
```

উদাহরণ ২: PL/SQL এর সাহায্যে আমরা ১ থেকে 10 পর্যন্ত লুপ চলাব, এবং প্রতিবারে ইনক্রিমেন্টেড ভ্যালুর মান জোড় না বিজোড় তা প্রিণ্ট করে দেখব।

PL/SQL কোড:

```
Declare
x number := 100;
Begin
for i in 1..10 loop
if mod(i,2) = 0 then
insert into temp values (i, x, 'i is even');
else
insert into temp values (i, x, 'i is odd');
end if;
x := x + 100;
end loop;
commit;
End;
```

আউটপুট:

1	100	i is odd
2	200	i is even
3	300	i is odd
4	400	i is even
5	400	i is odd
6	600	i is even
7	700	i is odd
8	800	i is even
9	900	i is odd
10	1000	i is even

উদাহরণ ৩. PL/SQL এর সাহায্যে যে কোন সংখ্যা Factorial বের করার program লিখুন।

Program:

DECLARE

```
fact NUMBER :=1;
```

```
n NUMBER := 5;
```

BEGIN

WHILE n > 0 loop

fact:=n*fact;

n:=n-1;

END LOOP;

dbms_output.put_line(fact);

END;

উদাহরণ ৪: We will retrieve the dname,location from dept table into 2 variable and then print them.

Solution:

Declare

```
v_deptno number(10);
v_dname varchar2(10);
v_loc varchar2(10);
```

Begin

```
v_deptno := &v_deptno;
Select dname,loc
Into v_dname,v_loc
From dept
Where deptno = v_deptno;
dbms_output.put_line('Dname is: ' || v_dname);
dbms_output.put_line('Location is: ' || v_loc);
End;
```

DML(Data Manipulating Language) in PL/SQL:

i. Insert

ii. Update

iii. Delete

ধরি নিচের মত একটি টেবিল আছে, এর উপর বিভিন্ন অপরিবেশন করতে হবে।

SQL> select * from student;

ID	SUB1	SUB2	SUB3
2	45	66	77
3	45	66	77
4	45	66	77
5	45	66	77
1	45	66	77

a. Inserting Example:

Declare

```
v_Id number(10);
v_Sub1 number(10);
v_Sub2 number(10);
v_Sub3 number(10);
```

Begin

v_Id := &v_Id;

v_Sub1 := &v_Sub1;

v_Sub2 := &v_Sub2;

v_Sub3 := &v_Sub3;

Insert into student

values(v_Id, v_Sub1, v_Sub2, v_Sub3);

dbms_output.put_line('Data Inserted');

End;

ii. Updating Example:

```
create table bank
(Acno number(10) primary key,
Balance number(20));
```

```
Declare
  v_acno number(10);
  v_inter number(10);
Begin
  v_acno := &v_acno;
  v_inter := &v_inter;
  Update bank
  Set balance = balance + v_inter
  where Acno = v_acno;
  dbms_output.put_line('Balance Updated');
End;
```

iii. Deleting Operation:

```
Declare
  v_acno number(10);
Begin
  v_acno := &v_acno;
  Delete from bank
  where acno = v_acno;
  dbms_output.put_line('Record Deleted');
End;
```

❖ Conditional Control in PL/SQL:

- 1) If...then
- 2) If...then..else
- 3) If...then..elseif

If...then Example: ধরি নিচের মত একটি টেবিল আছে, এর উপর কন্ট্রুক্ষন ব্যবহার করে আপডেট করতে হবে।

SQL> select * from bank;

ACNO	BALANCE
111	2200
222	333
333	4566
555	6777

```
Declare
  v_acno number(10);
  v_inter number(10);
```

```
Begin
  v_acno := &v_acno;
  v_inter := &v_inter;
  If v_inter < 500 then
    Update bank
    set balance = balance + v_inter
    where Acno = v_acno;
  End if;
```

```
end if;
End;
```

Output:

```
Enter value for v_acno: 111
old 8:   v_acno := &v_acno;
new 8:   v_acno := 111;
Enter value for v_inter: 200
old 9:   v_inter := &v_inter;
new 9:   v_inter := 200;
```

PL/SQL procedure successfully completed.

SQL> select * from bank;

ACNO	BALANCE
111	2200
222	333
333	4566
555	6777

If...then.....else Example:

```
Declare
  v_acno number(10);
  v_inter number(10);
Begin
  v_acno := &v_acno;
  v_inter := &v_inter;
  If v_inter < 500 then
    Update bank
    set balance = balance + v_inter
    where Acno = v_acno;
    dbms_output.put_line('Data updated');
  else
    dbms_output.put_line('No Data updated');
  end if;
End;
```

Assertions and Triggers

প্রশ্ন 1. Trigger কি? Trigger সর্বকে সংজ্ঞে দিলুন।
উত্তর: Database trigger হচ্ছে একটি procedural code, যা একটি table এ ব্যক্তিগতভাবে কার্যকর হয় বা যখন কিছু ঘটনা ঘটে তখন দেখা হয়। যখন নির্দিষ্ট শর্ত মিলে যায় তখন Trigger database এ সংরক্ষণ করা হয় এবং বাইবার কল করা হয়।

Example: Oracle, Microsoft, SQL Server, PostgreSQL, Firebird, MySQL/MariaDB, IBM DB2 LUW, SQLite, XML Database.

বিভিন্ন event সমূহের যে কোনটি execute করার জন্য Trigger দিয়ে হয়:

i. Database Manipulation Language (DML) statement (INSERT, DELETE, or UPDATE) execute করার জন্য।

ii. Database Definition Language (DDL) statement (CREATE, ALTER, or DELETE) execute করার জন্য।

iii. Database Operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN) execute করার জন্য।

Triggers are written to be executed in response to any of the following events.

- ❖ A database manipulation (DML) statement (DELETE, INSERT, or UPDATE).
- ❖ A database definition (DDL) statement (CREATE, ALTER, or DROP).
- ❖ A database operation (SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN).

These are the following advantages of Triggers:

- ❖ Trigger generates some derived column values automatically
- ❖ Enforces referential integrity
- ❖ Event logging and storing information on table access
- ❖ Auditing
- ❖ Synchronous replication of tables
- ❖ Imposing security authorizations
- ❖ Preventing invalid transactions

Trigger এর structure:

CREATE OR REPLACE TRIGGER trigger_name

BEFORE | AFTER | INSTEAD OF

INSERT | UPDATE | DELETE

OF col_name

ON table_name

REFERENCING OLD AS o NEW AS n

FOR EACH ROW

WHEN (condition)

DECLARE

Declaration-statements

BEGIN

Executable-statements

EXCEPTION

Exception-handling-statements

END;

BEFORE | AFTER | INSTEAD OF: আমাদের তৈরিকৃত Trigger কখন execute হবে, অপারেশনের আগে নাকি পরে তা এই Section এ উল্লেখ করতে হয়।

INSERT | UPDATE | DELETE: আমরা টেবিলের কোন Operation এর জন্য Trigger করতে চাই তা এই Section এ উল্লেখ করতে হয়। উল্লেখযোগ্য, টেবিলের প্রতিটি Operation এর জন্য Trigger করতে হয়। যেমন Insert এর জন্য Insert Trigger, Update এর জন্য Update Trigger এবং Delete এর জন্য Delete Trigger।

OF col_name ON table_name: ধরুন, ব্যাকের Accounts নামক টেবিলের Balance নামক একটি বো রয়েছে। আমরা চাই Balance জড়িত কোন ডাটা আপডেট হলে তার পূর্বে Balance Temp.Accounts নামক আরেটি বো হবে। একেতে OFF হবে Balance এবং ON হবে Accounts।

Declaration section: এই section টি DECLARE keyword দিয়ে আরও হয়। এটি একটি অপশনাল সেকশন। Program এ ব্যবহৃত সব ধরনের variable, cursor, subprogram এবং অন্যান্য উপাদান এই section এ define করা হয়।

Executable Section: এই section টি BEGIN keyword দিয়ে শুরু এবং END keyword দিয়ে শেষ হয়। এই section টি PL/SQL program এর mandatory Ask। এই section এ কমপক্ষে একটি কার্য নির্বাচন (executable) line থাকবে।

Exception Handling Section: এই section টি EXCEPTION keyword দিয়ে শুরু হয়। ইহাও একটি অপশনাল সেকশন, যা program এর error নির্বাচন করে।

উদাহারণ:

Accounts		
Acc_No	Transaction_id	Balance
101	TXN1201	20000
102	TXN1203	45100
103	TXN1205	100000
104	TXN1210	55000

উপরোক্ত টেবিলের জন্য আমরা একটি Trigger create করব, যা টেবিলের INSERT | UPDATE | DELETE উপর মন্তব্য করবে।

Trigger Creation:

Create Or Replace Trigger Accounts_changes
Before Delete Or Insert Or Update On Accounts
FOR EACH ROW WHEN (NEW.Acc_No > 0)

Declare

Balance_diff number;

Begin

sal_diff := :NEW.Balance - :OLD.Balance;

dbms_output.put_line('Old Balance:');

```

'||:OLD.Balance );
dbms_output.put_line(''NewBalance:''
||:NEW.Balance );
dbms_output.put_line(Balance difference:''
Balance_diff);
End;

```

উপরের কোডটি Execute করলে আউটপুট আসবে :
Trigger created.

১। এবার Accounts নামক টেবিলে আপারেশন চালানো যাব, আমরা Accounts এ Insert করব।
INSERT INTO Accounts
VALUES (105, TXN1211, 7500);

Output: Accounts_changes
Old Balance:
New Balance: 7500
Salary difference:

২। আমরা Acc_No104 এর Balance আপডেট করব।
UPDATE Accounts
SET Accounts = Accounts + 5000
WHERE Acc_No = 104;

Output: Accounts_changes
Old Balance: 55000
New Balance: 60000
Salary difference:5000

❖ Example:

Table 1:

Create table Bank
(Actno number(10) primary key,
Name varchar2(10),
Balance number(10))

• SQL> select * from bank;

ACNO	NAME	BALANCE
111	aaa	4000
222	bbb	300
333	ccc	3500

Table 2:

Create table Bank_Audit
(Acno number(10),
Name varchar2(10),
Old_Balance number(10),
Operation varchar2(10),
op_User varchar2(10),
op_date date);

• SQL> select * from bank_audit;

ACN	NAM	OLD_BALA	OPERATI	OP_US	OP_DA
222	bbb	300	update	SCOTT	25-JUN-04

Trigger Table1 & Table2 এর জন্য:

Operation:

Create Or Replace Trigger trig_bank After Update or Delete On Bank For Each Row Declare

```

v_oper varchar2(10);
v_Acno number(10);
v_Name varchar2(10);
v_balance number(10);

```

Begin

If Updating then

```

v_oper := 'update';
end if;

```

If Deleting then

```

v_oper := 'Delete';
end if;

```

```

v_Acno := :old.Actno;
v_Name := :old.Name;
v_balance := :Old.Balance;

```

Insert into bank_Audit
values(v_Acno,v_name,v_balance,v_oper,user,sy
date);

End;

• SQL> update bank

```

set balance = balance + 700
where actno = 222;

```

• SQL> delete from bank where actno = 111;

After Operation:

SQL> select * from bank_audit;

AC NO	NA ME	OLD_BAL ANCE	OPERA TION	OP_U SER	OP_D ATE
222	bbb	300	update	SCOTT	25-JUN-04
222	bbb	1000	update	SCOTT	25-JUN-04
111	aaa	4000	Delete	SCOTT	25-JUN-04

• SQL> select * from bank;

ACNO	NAME	BALANCE
222	bbb	1000
333	ccc	3500

যোজনাভূত DBMS এ নির্দিষ্ট শর্তপূরণ সাপেক্ষে কাজ করার জন্য হল ব্যবহৃত মেকানিজম Trigger। উদাহরণ ব্যৱপ বলা যায়। কোন যাকের ক্ষেত্রে যদি একজন গ্রাহক তার জমাকৃত টাকার চেয়ে বেশি টাকা ছাড়েন করে তবে তার banlance এ “-” চিহ্নযুক্ত হয়ে যাবে। এ ক্ষেত্রে যদি এই গ্রাহকের ব্যালেন্স শূন্য করে Overdraft ব্যবহার করতে পারবেন না। সুতরাং Assertion Balance column update করে বিবরিত রাখে।

Assertion: Assertion SQL এর একটি Statement যা Database একটি নির্দিষ্ট শর্ত নির্দিষ্ট করার জন্য সর্বদা বিদ্যমান থাকে। Assertion একটি টেবিলের Constraint হিসেবে কাজ করে। Assertion নির্দিষ্ট শর্ত একটি টেবিলে ভাটি Insert বা update থেকে বিবরিত রাখে। যেমন একজন Bank Account হোল্ডারের Balance ১০০ এর কম হলে তিনি তার Account withdrawal করতে পারবেন না। সুতরাং Assertion Balance column update করে বিবরিত রাখে।

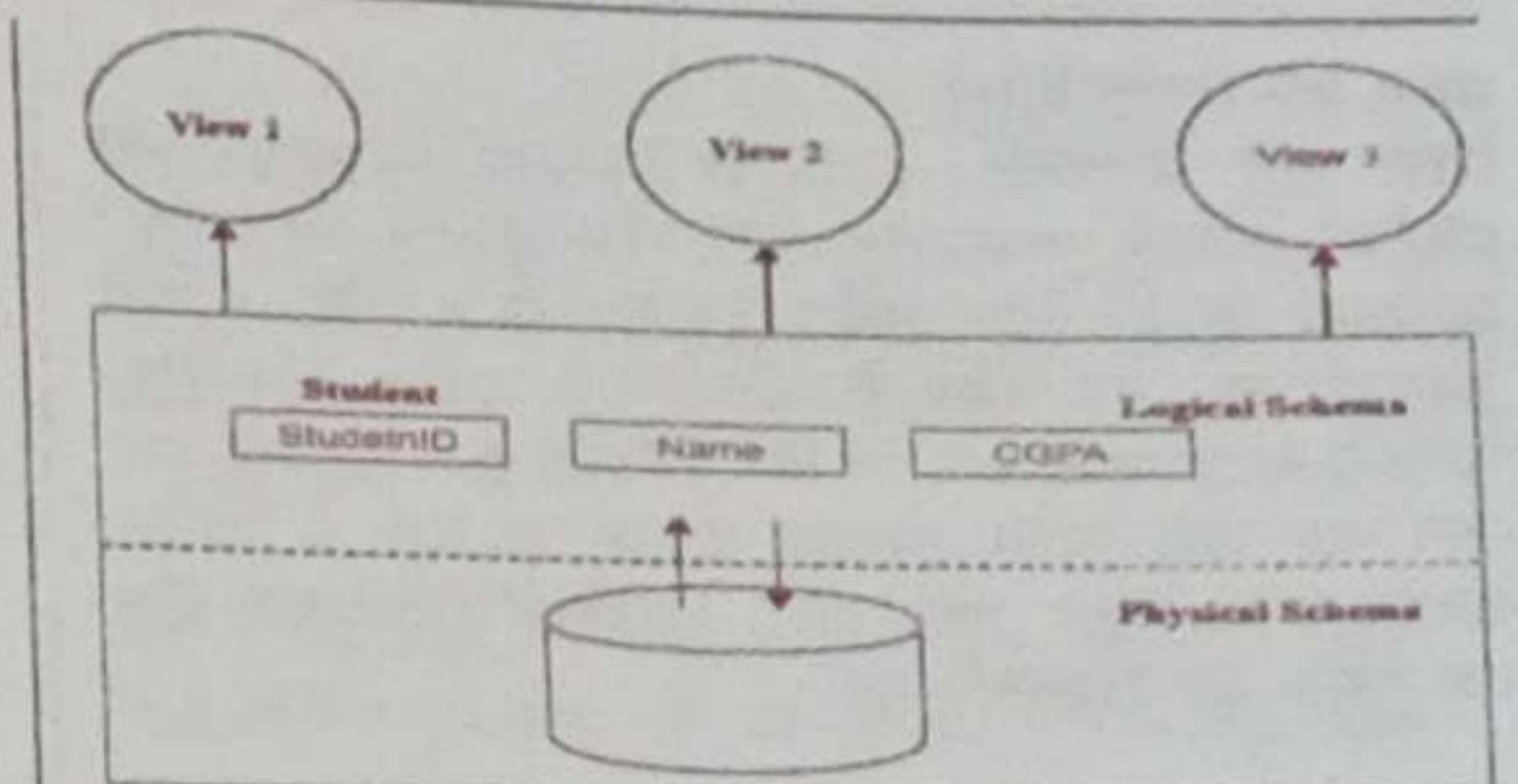


Fig: Database Schema

a) **Physical Schema:** Data Abstraction এর Physical level কে বর্ণনা করার জন্য যে Schema ব্যবহার করা হয় তাকে Physical Schema বলে।

b) **Logical Schema:** Data Abstraction এর Logical level কে বর্ণনা করার জন্য যে Schema ব্যবহার করা হয় তাকে Logical Schema বলে।

c) **Conceptual Schema:** Conceptual Schema হলো একটি মাধ্যম যা ডাটাবেস সিস্টেম সম্পর্কে পূর্ব পরিকল্পনা এবং তাদের বিলেখন সম্পর্কে Represent করে থাকে, তাকে Conceptual Schema বলে।

d) **Schema Object:** Logical Data Storage Structure কে Schema Object বলে।

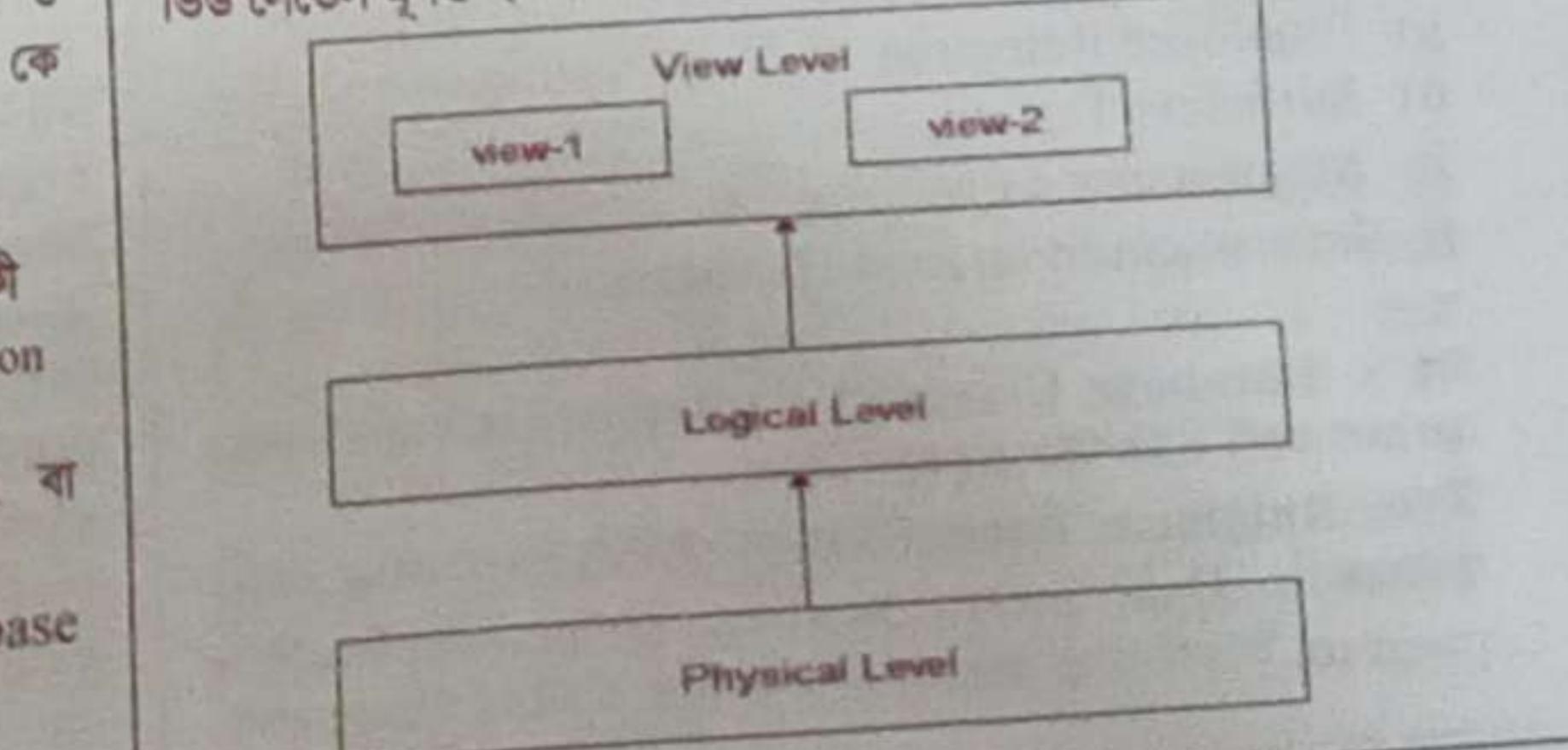
প্রশ্ন ৩. **Data Abstraction কি?** Data Abstraction এর বিভিন্ন স্তর (Levels of data Abstraction) এর বর্ণনা দাও।
উভয়ের ডাটা ভিত্তি এর ক্ষেত্রে জটিলতা পরিহার করে DBMS কে ইউজার এর নিকট Interactive করে তোলার জন্য প্রয়োজনীয় বৈশিষ্ট্যাবলী বাছাই করার প্রক্রিয়াকে ডাটা আবস্থাকরণ বলে।

বিভিন্ন Level এর বর্ণনা নিম্নরূপঃ

a) **Physical level:** এটি Data Abstraction এর সবচেয়ে Lowest level. Database এ কিভাবে ডাটা জমা হবে তা ডাটা আবস্থাকরণ এর যে লেভেলে বর্ণনা করা হয় তাকে ফিজিক্যাল লেভেল বলে।

b) **Logical level:** Data Abstraction এ কি ধরনের ডাটা জমা হবে এবং তাদের মধ্যে কি ধরনের বিলেখন থাকবে তা লজিক্যাল লেভেল এ বর্ণনা করা হয়।

c) **View level:** এটি Data Abstraction এর সবচেয়ে Highest level. একটি ডাটাবেস এর বিভিন্ন ভিত্তি সরবরাহ করাই এর উদ্দেশ্য। ভিত্তি লেভেল মূলত ইউজার এর সাথে interaction তৈরি করে।



প্রশ্ন ২. Database Schema কলাতে কী বৃুধ? ইহা কত প্রকার ও কী
বৃুধ Database Schema সমূহের বর্ণনা লিখ। Export Promotion
Bureau-2012

উত্তর: **Database Schema:** Database এর Overall বা
গুরুত্বপূর্ণ Design কে Database Schema বলে।
Database Schema ৪ প্রকার। নিম্নে ডাটাবেস জিমা (Database
Schema) সমূহের বর্ণনা দেওয়া হলোঃ

প্রশ্ন 4. ডাটা ডিকশনারি কি?

উত্তর: ডাটা ডিকশনারি -Data definition language এর সেট হচ্ছে তালোকে কল্পাইলেশন করলে একটি সেট অফ টেবিল পাওয়া যাব এনের কে এটি বিশেষ ফাইলে জমা করা হয় এ ফাইল কে ডাটা ডিকশনারি বলে। ডাটা ডিকশনারী মেটাডাটা (Metadata) ধারণ করে। মেটাডাটা বলতে আমরা ডাটা সম্পর্কিত ডাটাকেই বুঝি। সাধারণত এখন কোন নতুন ডাটা সিস্টেম প্রযোজন হয় তখন এ সিস্টেমে ব্যাবহৃত ডাটার বৈশিষ্ট্য, স্ক্যাপ, নাম, ইত্যাদি ভবিষ্যতের জন্য ধারণ করে রাখাকে ডাটা ডিকশনারি বলে।

প্রশ্ন 5. Modularity কি? BB-2017

উত্তর: Modularity- অলোগারিটি cardinality থেকে সম্পূর্ণ জালান। Modality indicate করে একটি entity এর instance অন্য একটি related entity এর instance সাথে ন্যূনতম কত বা যুক্ত হতে পারে তার সংখ্যা। Modality এর মান শূন্য বা এক হতে পারে।

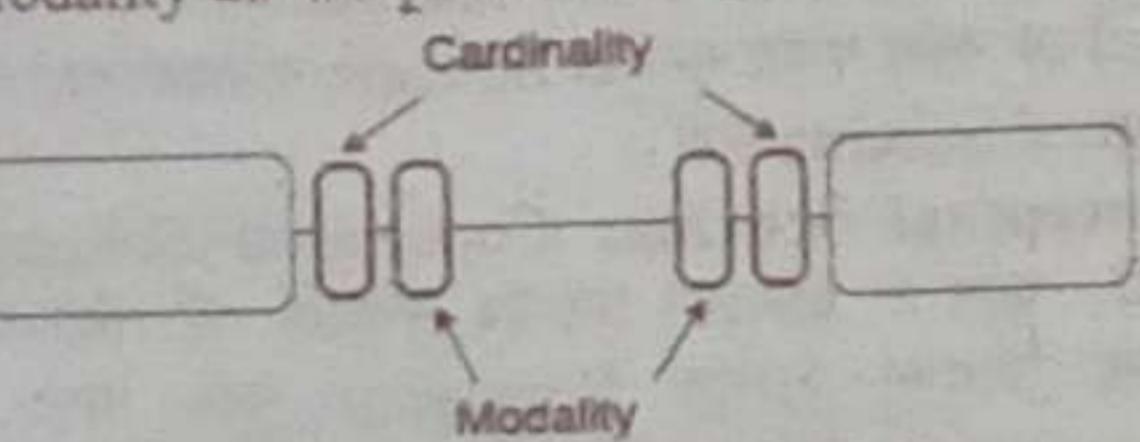


Fig: Cardinality and Modality

প্রশ্ন 6. What is ERP? What are the challenges of implementing ERP? [ERP কি? ইকারপি কর্তব্যান্বে চালোগারুলো কী?] [Janata Bank office (IT) -2015**]

উত্তর: Enterprise resource planning (ERP), এন এক ধরনের সংস্করণের বোকায় বা organization এর প্রতিবিনির কাবনারে ক্রিয়াকলাপ যেমন accounting, procurement, project management, risk management এবং supply chain operation পরিচালনা করতে ব্যবহার করে। একটি সম্পূর্ণ ERP suite এ enterprise performance management software অন্তর্ভুক্ত করা থাকে, যা পরিকল্পনা, বাজেট, পূর্বভাস এবং কোনও সংস্থার আর্থিক কলাকালের প্রতিবেদন করতে সহায়তা করে।

ERP কর্তব্যান্বে চালোগুলো:

- ERP Vendors
- Commitment from The Top Management
- Adequate Training
- Implementation Time
- Proper Project Management
- Implementation Cost
- Employee Retention
- Sufficient Testing
- Maintenance Cost
- Investment in Internal Hardware

প্রশ্ন 7. Database User/ডাটাবেস ইউজার কি? ডাটাবেস ইউজার এর ফল কী কি? সংক্ষেপে লিখুন।

উত্তর: Database User (ডাটাবেস ইউজার) যে সম্পত্তি ব্যক্তি ডাটাবেস এ Data insert, modify, create, application program ইত্যাদি কাজ করে থাকে, তাদেরকে ডাটাবেস ইউজার বলে।

অর্থাৎ যে সকল ব্যক্তি ডাটাবেস Manipulation করে, তাদেরকে ডাটাবেস ইউজার বলে।

ডাটাবেস ইউজারদের মোট চারটি গ্রন্থে ভাগ করা যায়। যথাঃ-

a) Application Programmer: কম্পিউটার প্রযোজন বৃন্দের বলা হয় Application Programmer।

b) Sophisticated user: যে সকল ইউজার সরাসরি সিস্টেম এর সাথে সরাসরি ইন্টারআক্টিভ (ডিজাশন) হয়, তাদেরকে Sophisticated user বলে।

c) Specialized user: যে সকল Sophisticated User Specialized Database এ প্রযোগ সৃষ্টি করে, তাদেরকে Specialized User বলে।

d) Naive user: ডাটাবেস এর সাধারণ ইউজারদের Naive user বলে। এনেরকে Unsophisticated User ও বলা হয়।

প্রশ্ন 8. Database Manager/ডাটাবেস ম্যানেজার কি? ডাটাবেস ম্যানেজার এর ফাংশন বা কাজ লিখ।

উত্তর: Database Manager/ডাটাবেস ম্যানেজার:

ডাটাবেস ম্যানেজার এর ফাংশন এবং Application Program এর মধ্যে Interface সরবরাহ করে এবং সিস্টেম কে Quary করে।

ডাটাবেস ম্যানেজার এর ফাংশন-

- File manager এর সাথে communication
- Integrity enforcement
- Security enforcement
- Backup & recovery
- Concurrency control

প্রশ্ন 9: Database Administrator কি? এর ফাংশন বা কাজ উল্লেখ কর। Ministry of Finance-2013

অবৰা, A company needs key person for DBMS. What is his/her duty as key person?

উত্তর: যে ব্যক্তি ডাটাবেস ডিজাইন, Manipulate পরিবর্তন, পরিবর্দন, পরিচালনা, নিরাপত্ত ইত্যাদি কার্যক্রমের দায়িত্ব পালন করে থাকে, সে ব্যক্তিকে ডাটাবেস Administrator বলে। A company key person is database administrator for DBMS.

Database Administrator এর ফাংশন বা কাজ/ Duty as key person:

- Schema definition
- Storage structure and access method definition
- Schema and physical organization modification
- Integrity constraint specification

প্রশ্ন 10. Information Processing Center এবং organizational চার্ট কি?

উত্তর: Information Processing Center: কেন organization এর যে বিভাগ ডাটা Entry Software উভয়, সিস্টেম এবং এলাইবিস ইত্যাদি কর্মকাণ্ড সম্পাদিত করে থাকে, তাদেরকে Information Processing Center বলে।

সংগঠনিক চার্ট: Information Processing Center এর যে বিভাগে কর্মসূচি কর্মকাণ্ড এবং কর্মচারীবৃন্দের হাইয়ার্কার্ড হলের মাধ্যমে প্রক্রিয়া করে থাকে, তাদেরকে সংগঠনিক চার্ট বলে।

অর্থাৎ যে সকল ব্যক্তি ডাটাবেস Manipulation করে, তাদেরকে

ডাটাবেস ইউজার বলে।

ডাটাবেস ইউজারদের মোট চারটি গ্রন্থে ভাগ করা যায়। যথাঃ-

a) DML Compiler

b) Embedded DML Pre-compiler

c) DDL interpreter

d) Query evaluation engine

১১. Database Security & Encryption

প্রশ্ন 1: ডেটাবেজ সিকিউরিটি কি? ডেটাবেজ সিকিউরিটি ক্ষেত্রে বিবেচ্য

বিষয় সমূহ আলোচনা করন। What is database security?

Discuss the issues to be considered in the field of database security.***

উত্তর: ডেটা এনক্রিপশন - যে প্রক্রিয়ায় ডেটাকে বিশেষ কোডের মাধ্যমে

পরিবর্তন করা হয় তাকে এনক্রিপশন বলে। Source ডেটাকে এনক্রিপ্ট

করে পাঠালে প্রাপক এ ডেটা ব্যবহারের পূর্বে decrypt করতে হয়।

এনক্রিপ্ট করা ডেটাকে নির্বিট নিয়মে মূল ডেটায় পরিবর্তন করাকে বলা

হয় decryption।

ডেটা এনক্রিপশনের প্রধান অংশ তাম্ব লিখুন। Write the main

parts of data encryption.***

উত্তর: ডেটা এনক্রিপশন - যে প্রক্রিয়ায় ডেটাকে বিশেষ কোডের মাধ্যমে

পরিবর্তন করা হয় তাকে এনক্রিপশন বলে। Source ডেটাকে এনক্রিপ্ট

করে পাঠালে প্রাপক এ ডেটা ব্যবহারের পূর্বে decrypt করতে হয়।

এনক্রিপ্ট করা ডেটাকে নির্বিট নিয়মে মূল ডেটায় পরিবর্তন করাকে বলা

হয় decryption।

ডেটা এনক্রিপশনের প্রধান চারটি অংশ-

(ক) Plain text

(খ) Cipher-text

(গ) Encryption algorithm

(ঘ) Security key or code

ডেটার সিকিউরিটি প্রধানত ২ ভাগে ভাগ করা যায়-

(ক) সিস্টেম সিকিউরিটি

(খ) ডেটা সিকিউরিটি

প্রেইন টেক্স্ট (Plain Text): এনক্রিপ্ট করার পূর্বে ডেটা যা পাঠ

করা যায় তাকে প্রেইনটেক্স্ট বলে। যেমন-'bitBox Job Solution'

সাইফারটেক্স্ট (Cipher-text): প্রেইন টেক্স্ট কে এনক্রিপ্ট করার

পরের বে ডেটা পাওয়া যায় তাকে সাইফারটেক্স্ট বলে। যেমন-

'jqbJwf Rwj Awtcqbqv' হল 'bitBox Job Solution' এর

সাইফারটেক্স্ট।

0 1 2 3 4 5 6 7 8 9 10 11 12

A B C D E F G H I J K L M

13 14 15 16 17 18 19 20 21 22 23 24 25

N O P Q R S T U V W X Y Z

[m= plaintext, k = ciphertext, উপরের টেবিল হতে, প্রতিটি

plaintext এর corresponding value এবং ciphertext এর

value নিয়ে total alphabet(26) বারা modulus করে যে সংখ্যা

পার তার corresponding letter হবে Cipher-text]

Example: bitBox Job Solution থেকে 'jqbJwf Rwj

Awtcqbqv'

সাইফারটেক্স্ট E=(m+k)mod 26

b = 1+8 mod 26 = 9 = j [যেখানে m = 1, k = 8]

i = 8+8 mod 26 = 16 = q [যেখানে m = 8, k = 8] এভাবে

বাকি textও পাওয়া যাবে।

এনক্রিপশন অ্যালগরিদম (Encryption Algorithm): যে

গাণিতিক ফর্মুলার মাধ্যমে প্রেইন টেক্স্ট থেকে সাইফার টেক্স্ট আবার

সাইফার টেক্স্ট থেকে প্রেইনটেক্স্ট এ ক্রস্পাস করা হয় তাকে

Encryption অ্যালগরিদম বলে।

সিকিউরিটি কী বা কোড (Security key or Code): যে কোডের

মাধ্যমে ডেটা Encryption & decrypt করা হয় তাকে সিকিউরিটি

কী বা কোড।

Basic SQL Query Example

Emp_No	Emp_Name	Salary	City
01	Hasan	30,000	Jamalpur
02	Habib	7,000	New York
03	Saiful	14,000	Gazipur
04	Amir	17,000	Khulna
05	Mahmud	15,000	New York

Table: Employee

□ Program 1. সকল Feild দেখার জন্য প্রয়োগ লিখ।
উত্তর: SELECT * FROM Employee;

□ Program 2. Employee Table হতে name ও Salary দেখার জন্য Program লিখ।
উত্তর: SELECT Emp_Name, Salary FROM Employee;

□ Program 3. যাদের নামের প্রথম অক্ষর A দিয়ে শুরু তাদের দেখার জন্য Program লিখ।
উত্তর: SELECT * FROM Employee
WHERE Emp_Name Like "A%";

□ Program 4. যাদের নামের শেষে A আছে তাদের বের করার জন্য Program লিখ।
উত্তর: SELECT * FROM Employee
WHERE Emp_Name Like "%A";

□ Program 5. যাদের নামের মাঝে A আছে তাদের বের করার জন্য Program লিখ।
উত্তর: SELECT * FROM Employee
WHERE Emp_Name Like "%A%";

□ Program 6. যাদের City New York তাদের বের করার জন্য Program লিখ।
উত্তর: SELECT * FROM Employee
WHERE City = "New York";

□ Program 7. যাদের Salary 10,000 টাকার উপরে তাদের বের করার জন্য Program লিখ।
উত্তর: SELECT * FROM Employee
WHERE Salary > 10000;

□ Program 8. যাদের Salary 10,000 টাকার কমতাদের দেখার জন্য Program লিখ।
উত্তর: SELECT * FROM Employee
WHERE Salary < 10000;

□ Program 9. যাদের Salary 10,000 থেকে 30,000 টাকার মধ্যে তাদের দেখার জন্য Program লিখ।
উত্তর: SELECT * FROM Employee
WHERE Salary BETWEEN 10000 AND 30000;

□ Program 10. যাদের Salary 20,000 টাকার কম তাদের Salary এর সাথে 1000 টাকা যোগ করে SQL Program লিখ।
উত্তর: SELECT Salary +1000
FROM Employee
WHERE Salary < 20000;

□ Program 11. যাদের Salary 10,000 টাকার কম এবং City New York তাদেরকে দেখার জন্য Program লিখ।
উত্তর: SELECT*
FROM Employee
WHERE Salary <10000 AND City = "New York";

□ Program 12. যাদের Salary 5,000 থেকে 15,000 এর মধ্যে নয় এবং নামের শেষে S নাই তাহলে তাদের বের করার জন্য Program লিখ।
উত্তর: SELECT * FROM Employee
WHERE Salary NOT BETWEEN 5000 AND 15000 AND Emp_Name NOT like "%S";

□ Program 13. যাদের Salary 5,000 থেকে 10,000 এর মধ্যে নয় এবং নামের শেষে Z নাই তাদের বের করার জন্য Program লিখ।
উত্তর: SELECT * FROM Employee
WHERE Salary NOT BETWEEN 5,000 AND 15,000 AND Emp_Name NOT like "%z";

□ Program 14. যাদের City New York তাদের Name, Salary ও City দেখাতে;
উত্তর: SELECT Emp_Name, Salary, City
FROM Employee
WHERE City = "New York";

□ Program 15. যাদের Salary 5,000 এর নিচে City New York তাদের Salary এর সাথে 2,000 টাকা যোগ করে দেখানোর জন্য Program লিখ।
উত্তর: SELECT Salary+2000
FROM Employee
WHERE Salary <5000 AND City = "New York";

□ Program 16. Employee Table হতে যার Emp_No 05 তার নাম update বা পরিবর্তন করে Mahmuda করার SQL Program টি লিখ।
উত্তর: UPDATE Employee
SET Emp_Name='Mahmuda'
WHERE Emp_NO=5;

□ Program 17. Employee Table হতে যার Emp_No 05 তার information delete করার SQL Program টি লিখ।
উত্তর: DELETE FROM Employee
WHERE Emp_NO=5;

□ Program 18. Employee Table নতুন একজন Employee Insert করার SQL Program টি লিখ।
উত্তর: INSERT INTO Employee Values(06, 'Rose', 20000, 'Berlin');

□ Program 19. Employee Table টি Delete করার SQL Program টি লিখ।
উত্তর: DELETE Table Employee;

Medium SQL Query Example

Employee নামক ডাটাবেসের টেবিল গুলো দেওয়া হল। উক্ত ডাটাবেস থেকে নিম্নর কুয়েরি গুলো সম্পর্ক করুন।

Employees

emp_id	emp_name	job_name	manager_id	hire_date	salary	commission	dept_id
68319	rkb Khan	President		1991-11-18	6000.00		1001
66928	Jony	Manager	68319	1991-05-01	2750.00		3001
65646	HR Rahman	Manager	68319	1991-04-02	2957.00		2001
69062	Sadik Khan	Analyst	65646	1991-12-03	3100.00		2001
64989	Rakib	Salesman	66928	1991-02-20	1700.00	400.00	3001
65271	Saifullah	Salesman	66928	1991-02-22	1350.00	600.00	3001
69324	Kazi	Clerk	67832	1992-01-23	1400.00		1001

Department

dep_id	dep_name	dep_location
1001	Finance	Dhaka
2001	Audit	Rajshahi
3001	Marketing	Sylhet
4001	Production	Chattogram

salary_grade

grade	min_sal	max_sal
1	800	1300
2	1301	1500
3	1501	2100
4	2101	3100
5	3101	9999

1. সকল manager কে SQL Command দেখার লিখুন।

উত্তর:

SELECT *
FROM employees e
WHERE emp_id IN
(SELECT manager_id
FROM employees);

2. সকল manager এর employee ID, name, job name, hire date এবং experience দেখার SQL Command লিখুন।

উত্তর:

SELECT *
FROM employees e
salary_grade s

উত্তর:
SELECT emp_id,
emp_name,
job_name,
hire_date,
age(CURRENT_DATE, hire_date) "Experience"
FROM employees
WHERE emp_id IN
(SELECT manager_id
FROM employees);

3. 2 এবং 3 grade এর সকল employees দেখার SQL Command লিখুন।

উত্তর:

SELECT *
FROM employees e
salary_grade s
WHERE e.salary BETWEEN s.min_sal AND
s.max_sal
AND s.grade IN (2, 3);

8. 4 এবং 5 grade এর সকল employees 4 and 5 দেখার SQL Command লিখুন যারা ANALYST এবং MANAGER হিসাবে কাজ করছেন।

উত্তর:

SELECT *
FROM employees e
salary_grade s

Database Management System

```

WHERE e.salary BETWEEN s.min_sal AND
s.max_sal
AND s.grade IN (4, 5)
AND e.emp_id IN
(SELECT e.emp_id
FROM employees e
WHERE e.job_name IN ('MANAGER',
'ANALYST'));

```

৫. এই সকল employees সকল দেখার SQL Command লিখুন
যাদের salary Jony এর salary থেকে বেশি।

```

SELECT *
FROM employees
WHERE salary >
(SELECT salary
FROM employees
WHERE emp_name = 'Jony');

```

৬. rkb Khan থেকে সিনিয়র সকল employees সকল দেখার
SQL Command লিখুন।

```

SELECT *
FROM employees
WHERE hire_date <
(SELECT hire_date
FROM employees
WHERE emp_name = 'rkb Khan');

```

৭. Write a query in SQL to list the employees
whose salary is same as the salary of Kazi or
Saifullah. List the result in descending order of
salary.

```

SELECT *
FROM employees
WHERE salary IN
(SELECT salary
FROM employees e
WHERE e.emp_name IN ('Kazi', 'Saifullah')
AND employees.emp_id <> e.emp_id);

```

৮. Write a query in SQL to list the employees
whose salary is more than the total remuneration
of the SALESMAN.

```

SELECT *
FROM employees
WHERE salary >
(SELECT max(salary+commission)

```

```

FROM employees
WHERE job_name = 'SALESMAN';

```

৯. Write a query in SQL to find the details of
highest paid employee.

```

SELECT *
FROM employees
WHERE salary IN
(SELECT max(salary)
FROM employees);

```

১০. Write a query in SQL to find the highest paid
employees in the department MARKETING.

```

SELECT *
FROM employees
WHERE salary IN
(SELECT max(salary)
FROM employees
WHERE dep_id IN
(SELECT d.dep_id
FROM department d
WHERE d.dep_name = 'MARKETING'));

```

১১. Write a query in SQL to list the details of the
senior employees as on year 1991.

```

SELECT *
FROM employees
WHERE hire_date IN
(SELECT min(hire_date)
FROM employees
WHERE to_char(hire_date,'YYYY') = '1991');

```

১২. Write a query in SQL to find the total salary
given to the MANAGER.

```

SELECT SUM (salary)
FROM employees
WHERE job_name = 'MANAGER';

```

১৩. Write a query in SQL to list the details of the
departments where maximum number of
employees are working.

```

SELECT *
FROM department
WHERE dep_id IN
(SELECT dep_id

```

Database Management System

```

FROM employees
GROUP BY dep_id
HAVING count(*) IN
(SELECT MAX (mycount)
FROM
(SELECT COUNT(*) mycount
FROM employees
GROUP BY dep_id) a));

```

১৪. Write a query in SQL to list the name, job
name, department name, location for those who
are working as a manager.

```

SELECT e.emp_name,
e.job_name,
d.dep_name,
d.dep_location
FROM employees e,
department d
WHERE e.dep_id = d.dep_id
AND e.emp_id IN
(SELECT manager_id
FROM employees);

```

১৫. Write a query in SQL to list the name of the
employees who are getting the highest salary of
each department.

```

Answer:  

SELECT e.emp_name, e.dep_id
FROM employees e
WHERE e.salary IN
(SELECT max(salary)
FROM employees
GROUP BY dep_id);

```

১৬. Write a query in SQL to find out the least 5
earners of the company.

```

SELECT *
FROM employees e
WHERE 5 >
(SELECT count(*)
FROM employees
WHERE e.salary > salary);

```

১৭. Write a query in SQL to list the employees
who joined in the company on the same date.

```

SELECT *
FROM employees e

```

```

WHERE hire_date IN
(SELECT hire_date
FROM employees
WHERE e.emp_id <> emp_id);

```

১৮. Write a query in SQL to list the department
where there are no employees.

```

SELECT b.dep_id,
count(a.dep_id)
FROM department b
LEFT OUTER JOIN employees a ON
a.dep_id=b.dep_id
GROUP BY b.dep_id
HAVING count(a.dep_id) = 0;

```

Advance SQL Query Example

1. Given two relations Employees (emp_name,
dept_id, salary), Department (dept_id,
dept_name), now update salary 10% increase
from Department table.

Answer:
UPDATE Employee
SET salary = salary + salary * .1
FROM Employee, Department
WHERE Employee.dept_id = Department.dept_id;

2. Write a SQL command to fetch first 3
characters name from employee.

Answer: [Help: substr ("expression", position,
length)
SELECT SUBSTR ("bitBox Group", 1, 3)
FROM Student;

3. How to delete duplicate student from Student
(std_id, name, gpa) table?

Answer:
DELETEFROM Student
WHERE std_id IN (SELECT *
FROM Student
GROUP BY std_id, name
HAVING COUNT (*) > 1)

How to remove duplicate rows in a SQL Server
table?

Answer: The primary key ensures that the table has
no duplicate rows. However, when you use the
SELECT statement to query a portion of the columns
in a table, you may get duplicates.

Database Management System

To remove duplicates from a result set, you use the DISTINCT operator in the SELECT clause as follows:

```
SELECT DISTINCT
    column1, column2, ...
    //Column Name
FROM
    table1; //Table Name
```

If you use one column after the DISTINCT operator, the database system uses that column to evaluate duplicate. In case you use two or more columns, the database system will use the combination of value in these columns for the duplication check. To remove the duplicates, the database system first sorts the result set by every column specified in the SELECT clause. It then scans the table from top to bottom to identify the duplicates that are next to each other. In case the result set is large, the sorting and scanning operations may reduce the performance of the query.

4. Consider the following database: "branch" table (branch_name, branch_city, assets) "customer" table (customer_name, customer_street, customer_city) "account" table (account_number, branch_name, balance) "loan" table (loan_number, branch_name, amount) "depositor" table (customer_name, account_number) "borrower" table (customer_name, loan_number)

Find all customers of the bank who have an account but not a loan in the bank.

Answer:

```
SELECT DISTINCT customer_name
    FROM depositor
    WHERE customer_name NOT IN(
        SELECT customer_name
        FROM borrower);
```

5. Find Nth Highest Salary of Employee?

Answer:

```
SELECT Id, Salary
    FROM Employee e1
    WHERE N - 1 = (SELECT
                    COUNT(DISTINCT e2.Salary)
                    FROM Employee e2
                    WHERE e2.Salary > e1.Salary)
```

EMP (emp_id, name, salary, city). Write a SQL to find the 3rd MAX salary in the EMP table?

Solution:

```
SELECT salary
```

```
FROM EMP
ORDER BY salary DESC
LIMIT 2, 1;
```

6. Find the name and salary of the employees whose salary is greater than the all departments.

Answer:

```
SELECT name, salary
    FROM Employees
    WHERE salary > ALL (
        SELECT MAX(salary)
        FROM Employees
        GROUP BY department_id);
```

7. Find all employees whose joining date between start to end. [01/10/2019 format]

Answer:

```
SELECT * FROM Employee
    WHERE joining_date BETWEEN
        (SELECT MIN(joining_date) FROM
            Employee)
    AND
        (SELECT MAX(joining_date) FROM
            Employee)
```

[Note: Date Format in SQL Server and MySQL is:
"MM/DD/YYYY"]

Date Format in Oracle:
to_date('31/01/2013','dd/mm/yyyy')]

8. Write a query to find distinct salary from employee table.

Answer:

```
SELECT DISTINCT salary
    FROM Employee e1 where e2 = (
        SELECT COUNT(DISTINCT Salary)
        FROM Employee e2
        WHERE e1.salary <= e2.salary);
```

Alternative Solution:

```
SELECT MIN(salary)
    FROM (SELECT DISTINCT salary FROM emp
    ORDER BY salary DESC) WHERE ROWNUM <= 2;
```

9. How to fetch monthly Salary of Employee if annual salary is given?

Answer:

```
SELECT Employee_name, Salary/12 as 'Monthly
    Salary' FROM EMPLOYEE;
```

10. What is the Query to fetch first record from Employee table?
Ans: SELECT * FROM Employee WHERE ROWNUM = 1;

11. What is the Query to fetch last record from the table?

Answer:
SELECT * FROM Employee WHERE Rowid = SELECT MAX(Rowid) FROM Employee;

12. What is Query to display first 5 Records from Employee table?

Answer:
Ans: SELECT * FROM Employee WHERE ROWNUM <= 5;

13. What is Query to display last 5 Records from Employee table?

Answer:
SELECT * FROM Employee e
 WHERE ROWNUM <= 5
 UNION
 SELECT * FROM
 (SELECT * FROM Employee e
 ORDER BY rowid DESC)
 WHERE ROWNUM <= 5;

14. How to Display Odd rows in Employee table?

Answer:
SELECT * FROM (SELECT ROWNUM AS rno,
 E.* FROM Employee E) WHERE MOD(rno,2) = 1;

15. How to Display Even rows in Employee table?

Answer:
SELECT * FROM (SELECT ROWNUM AS rno,
 E.* FROM Employee) WHERE MOD(rno,2) = 0;

Query to find out even number from given table -
Ans: SELECT *

```
    FROM table_name
    WHERE num % 2 = 0
```

16. How to fetch 3rd highest salary using Rank Function?

Ans: SELECT * FROM (SELECT Dense_Rank()
 OVER (ORDER BY salary DESC) as Rnk,E.*
 FROM Employee E) WHERE Rnk=3;

17. How Can I create table with same structure of Employee table?

Database Management System

Answer:
CREATE TABLE Employee2 as SELECT * FROM Employee;

18. Display first 50% records from Employee table?

Ans:
SQL Solution:
SELECT * FROM Employee
 LIMIT (SELECT COUNT(*)/2 FROM Employee)

Oracle Solution:
SELECT ROWNUM, e.* FROM emp e WHERE ROWNUM <= (SELECT COUNT(*)/2 FROM emp);

19. Display last 50% records from Employee table?

Answer:
SQL Solution:
SELECT * FROM Employee
 LIMIT (SELECT COUNT(*)/2 FROM Employee),
 (SELECT COUNT(*) FROM Employee)

Oracle Solution:
SELECT ROWNUM, E.* FROM Employee E
 Minus
 SELECT ROWNUM, E.* FROM Employee E
 WHERE ROWNUM <= (SELECT COUNT(*)/2 FROM Employee);

21. How can you fetch only common records between two tables?

Answer:
SELECT * FROM Employee;
 INTERSECT
 SELECT * FROM Employee1;

22. Find Query to get information of Employee where Employee is not assigned to the department.

Answer:
SELECT * FROM Employee WHERE Dept_no
 NOT IN (SELECT Department_no FROM Department);

23. Select all records from Employee table whose name is 'Amit' and 'Pradnya'

Answer:
SELECT * FROM Employee WHERE Name
 IN ('Amit', 'Pradnya');

24. Select all records from Employee table where name not in 'Amit' and 'Pradnya'

Answer:

```
SELECT * FROM Employee WHERE name NOT IN ('Amit', 'Pradnya');
```

25. How to fetch all the records from Employee whose joining year is 2017?

Answer:

Oracle command
SELECT * FROM Employee WHERE To_char(Joining_date,'YYYY')='2017';

MS SQL Command:
SELECT * FROM Employee WHERE SUBSTR(CONVERT(varchar, Joining_date,103),7,4)='2017';

26. How Do you find all Employees with its managers? (Consider there is manager id also in Employee table)

Answer:

```
SELECT e.employee_name,m.employee_name
FROM Employee e,Employee m WHERE
e.Employee_id=m.Manager_id;
```

27. Write a SQL query to show the name of employees who have joined in 2016 and salary is greater than 10000?

Answer:

```
SELECT name FROM Employee WHERE
Hire_Date LIKE '2016%' AND salary>10000;
```

28. How to display following using query?

*

**

Answer: We cannot use dual table to display output given above. To display output, use any table. I am using Student table.

```
SELECT lpad('**', ROWNUM, '**') FROM
Student WHERE ROWNUM <4;
```

29. How to add the email validation using only one query?

Answer: User needs to use REGEXP_LIKE function for email validation.

```
SELECT email FROM Employee
WHERE NOT REGEXP_LIKE(email, '[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}', 'i');
```

30. How can we display 1 to 100 Numbers with query?

Answer: SELECT level
FROM dual CONNECT BY level <=100;
[Note: User needs to know the concept of Hierarchical queries. Click here to get concept of hierarchical queries]

31. How to Find the Joining date of Employee in YYYY-DAY-Date format.

Answer: SELECT first_name,
to_char(joining_date, 'YYYY') JoinYear
,to_char(joining_date, 'Mon'),
to_char(joining_date, 'dd') FROM employees;

32. What query to display even records from table? BTCL-2017

Answer: SELECT *
FROM Table_Name
WHERE id IN (
SELECT id
FROM Table_name
WHERE id%2 = 0);

33. What is the query from show only duplicates in student(sid,name,gpa) table. Agrani Bank-2017

Answer: SELECT *
FROM Student GROUP BY std_id, name HAVING
COUNT (*) > 1;

34. There is a table Student, student (id, name, gender, age, address, phone, department, cgpa). Write a SQL Query to find the maximum CGPA of student from each department. DESCO Asst. Engineer (CSE)-2018

Answer: SELECT *
FROM student
WHERE cgpa IN (
SELECT MAX(cgpa)
FROM student GROUP BY 'department');

35. There are given tables, customer (cust_id, name, sales_id), order (order_id, cust_id, date, sales_id) salesman (sales_id, commission). Write a SQL Query to find the maximum CGPA of student from each department. Palli sanchay bank Asst. Database Administrator-2018

a) Find the customer details for those salesmen get commission greater than 12% commission.

Answer: SELECT customer.name, customer.address
FROM customer JOIN salesman ON
customer.sales_id=salesman.sales_id WHERE
customer.commission >,12;

Similar, Write a SQL Query: Commission greater than 10%
Answer:
SELECT*
FROM Table_Name
WHERE Commission>0.10

b) Count the salesman by their order_id and date.

Answer: SELECT order_id, date, COUNT (*)

FROM order GROUP BY order_id, date;

36. Construct a table "CUSTOMER", Where customerID is primary key. [Jibon Bima corporation Asst. Programmer-18]

CustomerID	CustomerName	Address	PostCode
1	Alfreds Futterkiste	Obere Str. 57, Berlin	12209
2	Around the Horn	120 Hanover Sq.London	WA1 1DP

Answer: CREATE TABLE customer (
CustomerID int NOT NULL,
CustomerName varchar(100) NOT NULL,
Address varchar(150) NOT NULL,
PostCode varchar(10),
PRIMARY KEY (CustomerID)
);

36. There are given tables, instructor (instructor_id, name, dept_name, salary), course (course_id, course_name, credit). Now write the following queries. [Bangladesh Krishi Bank Asst. Programmer -2018]

a) Find the names of all instructors whose salary is greater than the salary of all instructors in the Biology department.

Answer:
SELECT name
FROM instructor
WHERE salary > ALL (SELECT salary FROM
instructor WHERE dept name = 'Biology');

b) Delete all the employees from instructor list whose salary is less than average salary.

Answer: DELETE FORM instructor

WHERE salary< (SELECT AVG (salary)
FROM instructor);

c) Give 5% bonus to all instructors whose salary is less than the average salary.

Answer: UPDATE instructor
SET salary = salary*1.05
WHERE salary< (SELECT AVG (salary)
FROM instructor);

37. What is the query to select TOP 10% rows from grade table? Combine bank (HBFC & BK)-2018

Answer: Oracle Solution:
SELECT TOP 10 PERCENT *
FROM grade
ORDER BY rollNo;

38. Write a SQL query to insert tuple in table. NESCO-2018

Answer: The INSERT INTO statement is used to insert tuple in a table.

INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);

39. Show the 2nd maximum salary from table. BCPS-2018, BB-2017

Answer: There are two ways to query the 2nd maximum entity for a relation or table. The easiest way is using LIMIT. The syntax of finding Nth height value is

"SELECT column_name FROM table_name
ORDER BY column_name DESC LIMIT N-1, 1".

Here N value of max. If we have to find 2nd maximum the N=2 so the SQL query will be.

SQL Command:
SELECT salary
FROM Employee ORDER BY salary DESC LIMIT
1, 1;

40. EmpDetails নামক টেবিল থেকে Department wise total Salary বের করার SQL কমান্ড লিখুন। BB-2017
Answer: SELECT DeptID, SUM(Salary) FROM
EmpDetails GROUP BY DeptID;

41. EmpDetails নামক টেবিল থেকে department wise maximum salary বের করার SQL কমান্ড লিখুন। BCPS-2018

Answer: SELECT DeptID, MAX(Salary) FROM
EmpDetails GROUP BY DeptID;

42. students(sid,name,gpa) নামক table একই নামের একাদিক ছাত্র রয়েছে। এই সমস্যা students শুধু করার SQL কমান্ড লিখুন যাদের নাম একাদিক বাব রয়েছে। Agrani Bank-2017
Answer: SELECT * FROM students GROUP BY name HAVING COUNT(*) > 1

43. Students(id, name, gender, age, address, phone, department, cgpa). নামক টেবিল থেকে department wise maximum CGPA বের করার SQL কমান্ড লিখুন। DESCO Asst. Engineer (CSE)-2018
Answer: SELECT * FROM Students WHERE cgpa IN (SELECT MAX(cgpa) FROM Students GROUP BY 'department');

44. নির্ধারিত কার্য সম্পাদনের জন্য SQL Command লিখুন।

i. Employee Database তৈরি করুন।
ii. উক্ত Database থেকে Employee সের Date of Birth (DOB) এর Ascending Order এ নতুন একটি view তৈরি করুন। এখানে উল্লেখ যে, Employee Database এর Entity এবং Attributes আপনি নিজের মত নির্ধারণ করতে পারেন, তবে অবশ্যই DOB entity থাকতে হবে।

উত্তরঃ i)

```
CREATE TABLE IF NOT EXISTS employees (
EMPLOYEE_ID decimal(6,0) NOT NULL PRIMARY KEY,
FIRST_NAME varchar(20) DEFAULT NULL,
LAST_NAME varchar(25) NOT NULL,
EMAIL varchar(25) NOT NULL,
PHONE_NUMBER varchar(20) DEFAULT NULL,
DATE_OF_BIRTH date NOT NULL,
SALARY decimal(8,2) DEFAULT NULL,
MANAGER_ID decimal(6,0) DEFAULT NULL,
DEPARTMENT_ID decimal(4,0) DEFAULT NULL,
FOREIGN KEY(DEPARTMENT_ID,MANAGER_ID)
REFERENCES
departments(DEPARTMENT_ID,MANAGER_ID)
)ENGINE=InnoDB;
```

ii) SELECT *
FROM employee
ORDER BY DATE_OF_BIRTH ASC

45. Write those sql command. Employee(ename, street, city), Works (ename, cname, salary, joindate), Company(cname, city), Manages (ename,mname).

i) find name, street ,city ,who work for first corporation bank and earn more than 30000.

Table 1	Table 2	Result
ASP.NET C#.NET SQL 2K5 JQUERY VB.NET PHP MVC JAVA SQL 2K12	PHP MVC C#.NET JQUERY VB.NET SQL 2K12	ASP.NET SQL 2K5 VB.NET JAVA

Query:
SELECT *
FROM Table1 FULL OUTER JOIN Table2 ON
Table1.name = Table2.name
WHERE Table1.id IS null
OR Table2.id IS null

47. How to copy from Parent table to Child Table with 1 column dividing into 3 different column.

WITH cte AS (
SELECT Name, ROW_NUMBER() OVER
(ORDER BY Name) - 1 rn
FROM yourTable
)

SELECT
MAX(CASE WHEN FLOOR(rn / 2) = 0
THEN Name END) AS Name1,
MAX(CASE WHEN FLOOR(rn / 2) = 1
THEN Name END) AS Name1,
MAX(CASE WHEN FLOOR(rn / 2) = 2
THEN Name END) AS Name3
FROM cte
GROUP BY
rn % 2
ORDER BY
rn % 2;

48. Queries from HR schema:

i. Display details of jobs that were done by any employee who is currently drawing more than 15000 of salary.

```
SELECT JH.*
FROM JOB_HISTORY JH JOIN
EMPLOYEES E ON (JH.EMPLOYEE_ID =
E.EMPLOYEE_ID)
WHERE SALARY > 15000
```

ii. Display department name, manager name, and salary of the manager for all managers whose experience is more than 5 years.

```
SELECT DEPARTMENT_NAME,
FIRST_NAME, SALARY
FROM DEPARTMENTS D JOIN
EMPLOYEES E ON
(D.MANAGER_ID=E.MANAGER_ID)
WHERE (SYSDATE-HIRE_DATE) / 365 >
5
```

iii. Display employee name if the employee joined before his manager.

```
SELECT FIRST_NAME FROM
EMPLOYEES E1 JOIN EMPLOYEES E2
ON
(E1.MANAGER_ID=E2.EMPLOYEE_ID)
WHERE E1.HIRE_DATE <
E2.HIRE_DATE
```

iv. Display employee name, job title for the jobs employee did in the past where the job was done less than six months.

```
SELECT FIRST_NAME, JOB_TITLE
FROM EMPLOYEES E JOIN
JOB_HISTORY JH ON
(JH.EMPLOYEE_ID = E.EMPLOYEE_ID)
JOIN JOBS J ON (JH.JOB_ID = J.JOB_ID)
WHERE
MONTHS_BETWEEN(END_DATE,START_DATE) < 6
```

v. Display employee name and country in which he is working.

```
SELECT FIRST_NAME,
COUNTRY_NAME FROM EMPLOYEES
JOIN DEPARTMENTS
USING(DEPARTMENT_ID)
JOIN LOCATIONS USING(
LOCATION_ID)
JOIN COUNTRIES USING (
COUNTRY_ID)
```

vi. Display department name, average salary and number of employees with commission within the department.

```
SELECT DEPARTMENT_NAME,
AVG(SALARY),
COUNT(COMMISSION_PCT)
FROM DEPARTMENTS JOIN
EMPLOYEES USING (DEPARTMENT_ID)
GROUP BY DEPARTMENT_NAME
```

49. Consider the bank database in figure 1, where the primary keys are underlined, write the following SQL query for this relational database.

Branch(branch_name, branch_city, assets)

Customer(customer_name, customer_street,

customer_city)

Loan(loan_number, branch_name, amount)

Borrower(customer_name, loan_number)

Account(account_number, branch_name, balance)

Depositor(customer_name, account_number)

Query: Find all customers of the bank who have an account but not a loan,

Solution:

```
SELECT customer_name FROM Depositor
EXCEPT
SELECT customer_name FROM Borrower
```

50. There is a student table: Student (s_id, name, gender, age, phone, department, cgpa) Write a

query to get each department maximum CGPA of student and which student get highest cgpa in maximum of each department.

Answer:

Query 1:

```
SELECT name, department, MAX(cgpa) AS [Max CGPA]
FROM Student
GROUP BY department
ORDER BY department
```

Query 2:

WITH CTE AS

```
SELECT Name, Row_Number() OVER (
PARTITION BY Department
ORDER BY cgpa DESC) AS Rank
FROM Student)
SELECT S_ID, Department
FROM CTE
WHERE Rank <= 2;
```

51. Given two tables are employee (id, name, salary, dept_id,) and department (dept_id, dept_name), write SQL to find max salary and average salary for specific dept.

Query:

```
SELECT dept_id, MAX(salary) AS max_salary,
AVG(salary) AS avg_salary
FROM employee
GROUP BY dept_id;
```

52. Write SQL command for update salary if the salary is greater than 20000 update the salary 20% otherwise update 10%.

Solution:

```
UPDATE EMP SET salary = CASE
When salary>20000
    Then salary+salary*.20
Else salary+salary*.10
End
```

53. Suppose table: transmit (id, name, city) and travel (id, type, account). Now write SQL command of max, min, and avg of account for data travel type "bus".

Solution:

```
SELECT MAX(account), MIN(account),
AVG(account) FROM transmit JOIN travel ON
transmit.id = travel.id AND travel.type = "bus"
GROUP BY city;
```

54. Write a SQL query to remove duplicate from the EmployeeSalary (EmpID, Project, Salary) without using temporary table?

Solution:

```
DELETE FROM EmployeeSalary
WHERE EmpID IN (
SELECT EmpID
FROM EmployeeSalary
GROUP BY Project, Salary
HAVING COUNT(*) > 1);
```

55. SQL ব্যবহার করে Student নামে একটি টেবিল তৈরি করুন। টেবিল std-id, std-name, std-address এবং GPA নামে চারটি field থাকবে। Student টেবিল হতে যে সকল ছাত্রের ফলাফল GPA-5 তারের সকল রেকর্ড দেখানোর জন্য SQL Query লিখুন। উভয় Student নামে একটি টেবিল:

```
CREATE TABLE Student (
Std_id int primary key
Std_name varchar(255)
Std_address varchar(255)
GPA float(24)
)
```

Ans: SELECT *
FROM Student
WHERE GPA= 5.00;

56. What is the Output of the following Query-

ID	Name
1	Dhaka
11	Electric
2	Supply
1	Company
1	Limited
0	Bangladesh

Select * from myTable where id>0 order by id DESC, Name DESC

Solution:

ID	Name
11	Electric
2	Supply
1	Limited
1	Dhaka
1	Company

57. Salesperson (ID, Name, Age, Salary), Customer (ID, Name, City, Industry, Type)

Orders (Number, order_date, Cust_id, salesperson_id, Amount). Given table above , find the following:

- a)The names of all salespeople that have an order with Samsonic.
- b)The names of all salespeople that do not have any order with Samsonic.
- c)The name of all salespeople that have 2 or more orders.

a)SELECT sp.Name
FROM Salesperson sp
JOIN Orders o ON sp.ID = o.salesperson_id
JOIN Customer c on c.ID = o.cust_id
WHERE c.Name = 'Samsonic';

b)SELECT sp.Name
FROM Salesperson sp
WHERE sp.ID NOT IN (
SELECT o.salesperson_id
FROM Orders o JOIN Customer c
ON o.cust_id = c.ID
WHERE c.Name == 'Samsonic');

c)SELECT sp.Name
FROM Salesperson sp
JOIN Orders o ON sp.ID = o.salesperson_id
GROUP BY o.salesperson_id
HAVING COUNT(*) >= 2;

58. Tables: Actor(actor_id, name, year_born) Play(play_id, title, author, year_written) Role(actor_id, character_name, play_id). Now Write a SQL query that returns the number of actors who have performed in three or more different plays written by the author "August Wilson".

Answer:

```
SELECT count(r.actor_id)
FROM Role r, Play p
WHERE p.author = 'August'Wilson' AND
p.play_id = r.play_id
GROUP BY r.actor_id
HAVING count(DISTINCT r.play_id)>2
```

59. How to get distinct records from the Employees table without using distinct keyword.

Answer:

SELECT *

FROM Employees a
WHERE rowid = (SELECT max(rowid) FROM
Employees b WHERE a.Employees_no =
b.Employees_no);

60. Write an SQL query to print the name of employees having the highest salary in each department.

Answer:

```
SELECT empName,empDept,EmpSalary
FROM Employee
WHERE empSalary IN (SELECT
max(empSalary) AS salary FROM Employee
GROUP BY EmpDept)
```

61. Write a SQL query to fetch all the Employees who are managers who manage more than 6 employees from Employee table. Employee (EmpId,FullName,ManagerId)

Answer:

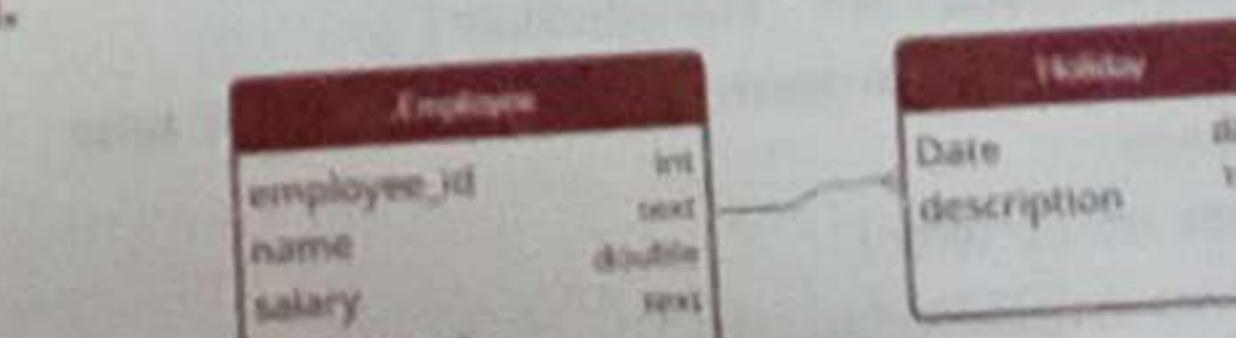
```
SELECT EmpId, FullName
FROM Employee
WHERE EmpId IN(
SELECT ManagerId
FROM Employee
GROUP BY ManagerId HAVING COUNT
(ManagerId)>6
)
```

62. Below tables are given,
Employee (employee_id, name, salary, department)
Leave (employee_id, date, reason, no_leaves)
Holiday (Date, description)

- i. Write mapping cardinality between 'Employee' and 'Holiday' table.
- ii. write query to show all employee's leave count.
- iii. Write query to show employees who are in 'HR' department and have taken at least 5 leaves.

Ans:

i.



ii. SELECT employee_id, COUNT (employee_id)
FROM Leave
GROUP by employee_id;

iii. SELECT *
FROM employee

```
WHERE employee_id IN (
    SELECT employee_id
    FROM leave
    GROUP BY employee_id
    HAVING COUNT(employee_id)>5);
```

63. What happened after executing the following SQL statement?

```
CREATE TABLE t (
    Val INT;
    INSERT INTO t(val) VALUES (1,2,2,3,
        "null", "null",4,5);
    SELECT COUNT(val)
    FROM t;
    SELECT COUNT DISTINCT(val)
    FROM t;
```

Ans:

1. Create table name "t" with one attribute name "val", which type is integer.
2. Insert value into "val" attribute.
3. 6 [Total number of value 1,2,2,3,4,5 = 6]
4. 5 [Unique value 1,2,3,4,5]

Others Related Questions

1. How to import data from .dump file in Oracle.

Answer: Import Command in Oracle:

```
Imp <username>/<password>@<hostname>
file=<filename>.dmp log=<filename>.log
full=y;
```

Export Command in Oracle:

```
expuserid /password@tnsname file=path.dmp
log=log.dmp full=y
```

2. Create user and password of db. Create index?

Answer:

```
Mysql>CREATE USER 'user1'@'local host'
IDENTIFIED BY 'Bangladesh1';
```

The above command will create a new user whose:

User name:user1

Password:Bangladesh1

To give permission to user1 we need to write :

```
Mysql>GRANT ALL PRIVILEGES ON
database_name.* TO 'user1'@'localhost' WITH
GRANT OPTION;
```

CREATE INDEX Syntax:

```
CREATE INDEX index_name
ON table_name(column1,column2,...);
```

Example:

```
CREATE INDEX custcity_country
ON customer(cust_city,cust_country);
```

Customer is the table name and cust_city, cust_country is column name of this table

3. In a relational database, a weak entity is an entity that cannot be uniquely identified by its attributes alone; Weak entity is dependent on strong entity and cannot exists without a corresponding strong. Now find the week entity set among the following entity set. [BCC(AP)-2021. BUET]

Student (studentid, name, surname)

Address (roadNo, postName, circle, personid, professor_ID)

Professor (professor_ID, professor_Name, professor_Salary)

course (courseID, subjectName, credit)

Ans: The weak entity in DBMS do not have a primary key and are dependent on the parent entity. It mainly depends on other entities. So, **Address table is a weak entity set.**

Software Engineering

[Syllabus: For BPSC CS: Introduction, Software Process. Project Management. Requirements Engineering Processes. System Models: Context Data, Behavioral and Object Models. Object Oriented Design Techniques. Real-Time Software Design, System Design with Reuse. Critical System Design Dependability, Software Maintenance, Critical System Specification and Development Verification and Validation. Software testing, Software Cost Estimation: COCOMO Model Halstead Formula.

Graph: Cel Analysis of Complexity Measures, Software Reliability and Availability, Quality Assurance.

For NTRCA CS: Paradigms, Requirements Analysis Fundamentals, Software Design Fundamentals, Software Testing Techniques and Strategies, Software Management and Maintenance Technique, CASE.

For NTRCA ICT: History, Nature, relation of software engineering to other discipline, Software Development Life Cycle, Programming language; Software nature and qualities: Product Qualities, Project Qualities, Correctness, Robustness, Usability, Maintainability, Portability, Quality Measurements; Software development life cycle: Requirement, Design, Development, Testing, Maintenance; Software development model: Waterfall, Agile, Spiral, RDD, V-Model; Software Engineering Principles: Modularity, Abstraction, Generality, Object Oriented, Component Oriented, Structured.; Specification and Verification: Requirement Specification, Descriptive Specification, Testing, Analysis, Debugging; Modeling and Design: basics of modeling diagram, UI design. Software Project Management: Concepts, Project Metrics, Estimation, Risks Management.]