Page No.
Date

Name: Liza Lad Enrollment Number: 200303108
Subject: DBMS Code:

# Ch-2
# Relational Query Languages

## Multiple Choice Questions:

~~(1.)~~

(1.)

(2.)

(3.) (d) DML

(4.) (a) Delete records from table.

(5.) (d) All of the above

(6.) (c)

(7.) (c)

(8.)

(9.) (d)

(10.) (b)

(11.) (a)

## Fill in the blanks:

(1.) procedural query language

(2.) Five Basic

(3.) a binary operator

(4.) Rename operator

(5.) Non procedural query language

(6.) two

(7.)

(8.) Data Query Language

(9.)

(10.) DML

## Short Answer questions:

(1.) What is relational algebra?

Sol$^n$ The relational algebra is a procedural query language, serves as the basis for the SQL language.
It consist of a set of operations that take one or two relations as input and produce a new relation as their result.

Q.2 Wha

Sol$^n$ Rela que algeb eal

Q.3 wh ope

Sol$^n$ It re u a

Q.4

Sol$^n$

Q.2 What is Relational calculus?

Sol$^n$ Relational calculus is a non-procedural query language, and instead of algebra, it uses mathematical predicate calculus.

Q.3 What are the fundamental relational operators?

Sol$^n$ The fundamental operations in the relational algebra are select, project, Union, set difference, Cartesian product and rename.

Q.4 Explain tuple and domain relational calculus.

Sol$^n$ Tuple relational calculus.

It is a non-procedural query language. It describes the desired information without giving a specific procedure for obtaining that information.

Domain relational calculus.

It uses domain variables that take on values from an attributes domain, rather than values for an entire tuple.

Q-5 What is SQL 3? Answer in brief.

Sol^n SQL 3 includes data definition and management techniques from Object-Oriented dbms, OO-dbms, while maintaining the relational dbms platform.

Q-6 Explain DDL and give its examples.

Sol^n The SQL DDL provides commands for defining relation schemas, deleting relations, and modifying relation schemas.

Examples
CREATE: To create objects in a database

ALTER: To alter the schema, or logical structure of the database.

DROP: To delete objects from the database

TRUNCATE: To remove all records from the table.

Q-7 Explain DML and give its examples.

Sol^n It is a set of SQL commands used to insert, modify and delete data in a database.
It is normally used by general users who are accessing database via pre-developed

Examples:
Insert: to insert data into a table.

Update: to modify existing data in a table

Delete: to delete records from a table

Page No.
Date

Q8 How idomain and tuple relational calculus different.

Sol$^{n}$

| Domain Relational Calculus | Tuple Relational Calculus |
|---|---|
| → The variables represent the value drawn from specified domain. | → The variables represent the tuples from specified relation |
| → A domain is equivalent to column data type and any constraints on value of data | → A tuple is a single element of relation. In database term, it is a row. |
| → In this filtering is done based on the domain of attributes. | → In this filtering variables uses tuple of relation. |
| → Notation: $\{a_1, a_2, a_3, \ldots, a_n \mid P(a_1, a_2, a_3, \ldots, a_n)\}$ | → Notation: $\{T \mid P(T)\}$ or $\{T \mid \text{Condition}(T)\}$ |
| → Example: $\{ \mid < \text{Employee} > \text{Dept\_Id} = 10 \}$ | → Example: $\{T \mid \text{Employee}(T) \text{ and } T.\text{Dept\_ID} = 10\}$ |

Page No.
Date

Q9 Explain unary relational operations.

Sol" A unary operator is an operator that operates on only one operand.

As unary operations have only one operand they are evaluated before other operations containing them.

Q10 Explain binary relational operations.

Sol" A relational expression consists of two operands separated by a relational operator.
If the relation is satisfied, it has the value true. ~~The result of a relational operation is therefore a Boolean value.~~
If the relation is not satisfied, it has the value false.

## Long Answer Questions:

**Q1** Differentiate between DML and DDL

**Sol^n**

| | DDL | DML |
|---|---|---|
| → | A type of SQL command that helps to define database schemas | A type of SQL command that helps to retrieve and manage data in relational databases. |
| → | Stands for Data Definition Language | Stands for Data manipulation language. |
| → | Create, drop, alter are some DDL commands | Insert, update, delete and select are some commands. |
| → | Commands affect the entire database or the table | Commands affect one or more records in a table. |
| → | SQL statements cannot be rolled back. | SQL statements can be rolled back. |

Page No.
Date

Q.2 Explain SQL3 and its features.

Sol": SQL3 is a superset of SQL/92, in that it supports all of the constructs supported by that standard, as well as adding new ones of its own.

Features of SQL3

(1) Classification hierarchies.

(2) Embedded structures that support composite attributes.

(3) Collection data-types that can be used for multi-valued attribute types.

(4) Large Objects types, LOBs, within the DB, as opposed to requiring external storage.

(5) User defined data-types and functions that can be used to define complex structures and derived attribute value calculations, among many other function extensions.

Page No.
Date

Q3 Explain both the types of relational calculus and give the difference between the two.

Sol$^{n}$

## Relational Calculus

### Tuple Relational Calculus (TRC)

It is specified to select the tuples in a relation.

In TRC, filtering variable uses the tuples of a relation

The result of the relation can have one or more tuples

Notation:

$\{T \mid P(T)\}$ or $\{T \mid \text{Condition}(T)\}$

where T is resulting Tuples

P(T) is the condition used to fetch T.

### Domain Relational Calculus (DRC)

It uses the same operators as tuple calculus. It uses logical connectives ∧ (and), ∨ (or)

In DRC, filtering variable uses the domain of attributes

It uses Existential (∃) and universal Quantified (∀) to bind the variables.

Notation:

$\{a_1, a_2, a_3, \ldots, a_n \mid P(a_1, a_2, a_3, \ldots, a_n)\}$

$a_1, a_2$ are attributes

P stands for formula built by inner attributes.

Page No.
Date

Q.4 Explain relational algebra and ~~give examples~~ explain its fundamental types with examples.

Sol^n The relational algebra is a procedural query language. Serves as the basis for the SQL language.
It consists of a set of operations that take one or two relations as input and produce a new relation as their result.

## Fundamental types.

The fundamental operations in the relational algebra are:
Select ($\sigma$)
project ($\pi$)
Union ($\cup$)
Set difference ($-$)
Cartesian product ($\times$)
Rename ($\rho$)

Q.5 Explain SQL and its types of commands

Sol" SQL (Structured Query language) is a standardized programming language that is used to manage relational databases and perform various operations on the data in them.

## Types of SQL Commands

DDL :- Data Definition language
DQL :- Data Query Language
DML :- Data Manipulation language
DCL :- Data Control Language

# Relational Database Design.

## Multiple Choice Questions

(1.) (b)

(2.)

(3.) (b)

(4.) (d.)

(5.) (b)

(6.) (c)

(7.) $B \rightarrow D$

(8.) (a)

(9.) (d)

(10) (c)

## Fill in the Blanks

(1) Data Redundancy and insertion

(2) Trivial

(3)

(4) 3 NF

(5) Key

Page No.
Date

## Short Question

**Q.1.** What is functional dependency? Explain trivial and non trivial functional dependency with example.

**Sol$^n$** **Functional dependency :-**

It is a constraint that specifies the relationship between two sets of attributes where one set can accurately determine the value of other sets.
It is denoted as $X \rightarrow Y$, where X is a set of attributes that is capable of determining the value of Y.

**Trivial functional dependency :-**

If a functional dependency (FD) $X \rightarrow Y$ holds, where Y is a subset of X, then it is called a trivial FD.
Trivial FD's always hold

**Non-trivial functional dependency :-**

If an FD $X \rightarrow Y$ holds, where Y is not a subset of X, then it is called a non-trivial FD.

## Long Questions:

Q1 What is meant by normalization? Write its need. List and discuss database anomaly during database design.

Sol" **Normalization:-**

It is used to minimize the redundancy from a relation or a set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.

**Need of Normalization:-**

- Eliminates redundant data
- Reduces chances of data errors
- Reduces disk space
- Improve data integrity, scalability and data consistency.

There are three types of anomalies: update, deletion and insertion anomalies.

Page No.
Date

tion 37

Q-2 Consider schema Employee(E-ID, E-NAME, E-CITY, E-STATE) and
FD = { E-ID → E-NAME, E-ID → E-CITY, E-ID → E-STATE, E-CITY → E-STATE }
(1) Find attribute closure for : (E-ID)*
(2) Find (E-Name)*

Sol" Given functional Dependency :-
{ E-ID → E-Name, E-ID → E-CITY, E-ID → E-STATE, E-CITY → E-STATE }

(1.) Closure for : (E-ID)*

E-ID → E-ID, E-STATE, E-CITY, E-NAME

(2) Closure for : (E-Name)*

E-Name → E-Name.

Page No.

Page No.
Date

# Query processing and Query Optimization

## Multiple Choice Questions:

(1)

(2)

(3)

(4)

(5)

(6)

(7)

## Long Questions

Q.1) Explain evaluation expression process in query optimization

Sol$^n$ ~~In this method, the given~~.
To evaluate an algebric expression means to find the value of the expression when the variable is replaced by a given number. To evaluate an expression, we substitute the given number ~~of~~ for the variable in the expression and then simplify the expression using the order of operations.

Q-2 Explain steps in Query processing. [4]

Sol$^n$ (1) Scanning and Parsing

Scanning is the process of converting the query text into a tokenized representation

The parser checks the tokenized representation for correct syntax.

(2) Query Optimization or planning the execution strategy

The goal of the query optimizer is to find a resonably efficient strategy for executing the query using the access routines.

(3) Query Code Generator

Once the query optimizer has determined the execution plan the code generator writes out the actual access routines to be executed.
With an interactive session, the query code is interpreted and passed directly to the runtime database processor for execution.
It is also possible to compile the access routines and store them for later execution

(4.) Execution in the runtime database processor

At this point, the query has been scanned, parsed, planned and compiled.
The runtime database processor then executes the across routines against the database.
The results are returned to the application that made the query in the first place.
Any runtime errors are also returned.