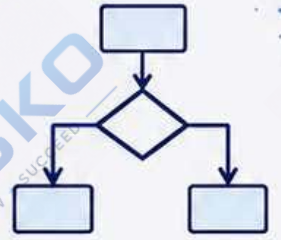


1010101010
0101010101
1010101010
0101010101



Notes





Unit - II

any one
6 marks
from two.

classmate
Date _____
Page _____

Q4.* What are constraints? Explain Primary, key Foreign key, Unique, Check, Not NULL with example.

→

Constraints -

Constraints are rules applied on table columns to maintain accuracy and consistency of data in a database. They prevent invalid data from being inserted, updated, or deleted.

Constraints help enforce business rules at the database level.

Common constraints are -
Primary key, Foreign key, Unique, Check and Not NULL.

1. Primary Key Constraint -

A primary key uniquely identifies each row in a table.

→ It cannot contain NULL values.

→ It must contain unique values.

→ A table can have only one primary key.

Eg- CREATE TABLE Students (
RollNo INT PRIMARY KEY,
Name VARCHAR (30),
Age INT
);

Here, RollNo is the primary key.

No two students can have the same RollNo and it can not be left empty.

2. Foreign Key Constraint

A Foreign key creates a relationship between two tables.

It refers to the primary key of another table.

It ensures referential integrity, meaning you cannot insert a value that does not exist in the parent table.

Eg-

```
CREATE TABLE Marks (
  RollNo INT,
  Subject VARCHAR(20),
  Marks INT,
  FOREIGN KEY (RollNo)
  REFERENCES students (RollNo)
);
```

Here, RollNo in the Marks table must

match an existing RollNo in the students table.

You cannot insert marks for a student who does not exist.

classmate

Date _____
Page _____

3. Unique Constraint -

A Unique constraint ensures all values in a column are different. / Unique. It allows one NULL value but no duplicates.

Eg-

```
CREATE TABLE Employee
```

```
C
```

```
EmpID INT PRIMARY KEY,
```

```
Email VARCHAR(50) UNIQUE,
```

```
Mobile VARCHAR(45)
```

```
);
```

Here, Email must be unique for every employee.

4. Check Constraint -

A Check constraint restricts the values that can be stored in a column. It ensures only valid data is entered according to a condition.

Eg- CREATE TABLE Product C

```
ProductID INT PRIMARY KEY,
```

```
Price INT CHECK (Price > 0),
```

```
Quantity INT CHECK  
(Quantity >= 0)
```

Price must be greater than 0 and quantity cannot be negative.

4. NOT NULL Constraint-

Not Null ensures that a column cannot be left empty. It forces the user to enter a value.

Eg-

CREATE TABLE Customer

CustID INT PRIMARY KEY,

Name VARCHAR(30) NOT NULL,

City VARCHAR(30)

Here, Name must be always have a value, it cannot be NULL.

Conclusion-

Constraints enforce rules on data and help to maintain reliability, consistency and accuracy in a database.

Q2.★ What is View? Explain its need, advantages and types with examples.

→ View-

A view is a virtual table created from one or more existing tables.

It does not store data physically. It displays data that is fetched from base tables using a SELECT query.

A view behaves like a table while retrieving data, but the data inside it is not stored separately.

When the base table changes, the view reflects the updated data.

Syntax-

```
CREATE VIEW view_name AS  
SELECT columns  
FROM table_name  
WHERE conditions;
```

Advantages of View-

1) Improves Data Security-

A view can hide sensitive columns and give access only to required information. Users see restricted data while the base table remains protected.

2) Simplifies Complex Queries-

Complicated joins and calculations can be stored inside a view. User access them with a simple

`SELECT * FROM view-name;`

3) Provides Logical Data Independence-

If the structure of the base table changes, the view can still present data in the same format.

4) No Extra Storage Required-

Views do not store data physically, so there is no additional space consumption. Data is always fetched from base tables.

Disadvantages of View-

1) Some views are not updated. updatable.

Complex view that included joins, group functions or multiple tables cannot be updated.

Only simple views are usually updated.

2) Slower Performance.

Because views fetch data from base tables every time, complex views may causes slower execution.

3) No Indexing on View-

Indexes cannot be created directly on normal views.

4) Increase Dependancy-

If the base table structure changes too much, some views may become invalid and need to be recreated.

classmate

Date _____
Page _____

Q3. Explain different types of joins with examples; Inner, Outer, Left, Right, Full, Cross.



Join - It is used to fetch data from two or more tables.

A join is an operation in SQL that combines rows from two or more tables based on a related column between them.

⇒ Inner Join -

Inner join returns only the matching rows from both tables. Rows that do not match are not shown.

Q4. * Write short notes on in-built functions - Character, Number, Date, Group, and Conversion functions -

SQL provides several built-in function that perform operations on data.

These functions make calculations, modify text, format dates, and summarize group of rows.

The important categories are character Number, Date, Group and conversion function.

i) Character (String) Functions -

Character functions work on text or string values.

i) UPPER() -

Convert text to uppercase.

eg -

```
SELECT UPPER ('hello'); -- HELLO
```

ii) LOWER() -

Convert text into lower case.

eg -

```
SELECT LOWER ('HELLO'); -- hello
```

classmate

Date _____

Page _____

iii) **LENGTH ()** -

Returns the number of character

`SELECT LENGTH ('India'); -- 5.`

iv) **SUBSTR (string, start, length) -**

`SELECT SUBSTR ('DATABASE', 2, 3); -- ATA`

v) **INSTR (string, substring) -**

`SELECT INSTR ('Computer', 'put'); -- 4.`

vi) **CONCAT () -**

joins two string.

`SELECT CONCAT ('Data', 'base'); -- Database`

WL2SIC

2) Number Functions -

These Functions perform mathematical operations.

ARFCP

i) ABS () -

Return absolute value.

SELECT ABS (-15); -- 15

ii) ROUND (num, decimal)

Rounds the value.

SELECT ROUND (13.1468, 3); -- 13.146

iii) FLOOR (num) -

Returns the greatest integer less than or equal to the number.

SELECT FLOOR (12.34); -- 12

iv) CEIL (num) -

Returns the smallest integer greater than or equal to the number.

SELECT CEIL (12.1); -- 13

v) POWER (num, exponent) - Rises a number to the power

SELECT POWER (2, 3); -- 8

3. Group (Aggregat) Function -

Group function also known as aggregate functions, operates on a set of rows and return a single value for that set. They are commonly used for with GROUP BY clause to perform calculations on group of rows.

CSAMM

- i) COUNT() - It return the numbers of rows in a group.

```
SELECT COUNT(*) FROM student;
-- Count all the rows from stud
```

- ii) SUM() - Calculate the sum of numbers.

```
SELECT SUM (salary) FROM employee;
```

- iii) AVG() - Calculate the average value of a numeric ~~value~~ column.

```
SELECT AVG (Age) FROM students;
```

classmate

Date _____
Page _____

iv) MIN C) -

Returns the minimum values in a column.

```
SELECT MIN (Salary) FROM Employees;
```

v) MAX C) -

Returns the maximum value in a column.

```
SELECT MAX (Salary) FROM Employees;
```