SQL

Q. what is Distinct ? how to use it.
Q can we rename column of output,

Camlin | Page
Date / /

Select Statement

SELECT - used to retrieve the data

Statements	Clauses
1) SELECT UNIQUE (column)	i) WHERE - condition
SELECT DISTINCT (column)	ii) GROUP BY - for grouping & using aggregate functions.
- Retrieve only unique elements from column	iii) HAVING - It selects among the groups formed by GROUP BY clause
for aa bb c d ee	iv) ORDER BY - order the column in ASC. DESC
unique → c d	
Distinct → a b c d e	

2) SELECT COUNT (column name)
SELECT COUNT (DISTINCT columnname)
SELECT COUNT (*)
- Retrive number of rows can be used with GROUPBY or WHERE clause.

3) SELECT TOP (number)
FROM table
- Retrieve top given number of results.

4) SELECT FIRST (column), LAST (column2), ~~RANDOM (column3)~~
- Retrieve first, last, ~~and random~~ any value

5) SELECT TOP 1 (column)
FROM table
ORDER BY NEWID()
- Retrieve the random row in SQL server syntax vary to other systems.

6) SELECT column AS 'columnname
- column name changes from column to columnname. and column is selected.

Subselect

a	b
NY0123'	US4567
AZ9437	GB1234
CA1279	FR5678

```
SELECT  SUBSTR (a, 1, 2)  AS  State,
        SUBSTR (a, 3)     AS  Statecode,
        SUBSTR (6, 1, 2)  AS  Country,
        SUBSTR (6, 3)     AS  Countrycode
FROM  table1 ;
```

Output

State	Statecode	Country	Countrycode
NY	0123	US	4567
AZ	9437	GB	1234
CA	1279	FR	5678

Q. what are different clauses? Q. ... BETWEEN

Q. what is where clause?

Q. How to write query to return ... starting with 'K' letter?

Camlin Page Date

Clauses

1) WHERE - Gives condition for Data retrival using select statements hence narrowing data returned by query hence filtering.

Syntax

```
SELECT columnnames,....
FROM
WHERE (condition)
```

In condition we use operators like

1) logical ⇒ AND, OR - for many conditions
2) comparator ⇒ <, >, <=, >=, =, <> for not equal
3) LIKE operator ⇒ %a, a%, _a, a_, a%z, _a%

   eg. WHERE column LIKE '%a_';

   It gives results with 'a' letter at second last position in value.

* To find null we use

  WHERE column1 IS NOT NULL ✓

  WHERE column1 IS NULL

  WHERE column1 = ✗

4) BETWEEN ⇒ to find numbers in betn range

   WHERE columnname BETWEEN 10 AND 20

   WHERE columnname NOT BETWEEN 10 AND 20

   we can use BETWEEN operators for Integer, TEXT & DATETIME as well.

5) IN operator ⇒ multiple values in where clause

   WHERE columnname IN (value, value2, value3)

   WHERE columnname NOT IN (value1, value2, value...

   we can use subquery to give return values to IN operator.

Q. what is order by clause? How to use it
Q. what is Having clause? How to use it
Q. what is Using clause? How to use it

2) ORDER BY - To give aphabetical, numerical orders which can be ascending by default and descending if mentioned to columns specified.

Syntax

1)
```
SELECT *
FROM table
ORDER BY (column) ASC OR DESC;   ← ASC/DESC can be used.
```

2)
```
SELECT *
FROM table
ORDER BY (column) ASC
LIMIT num        ← Limit the number of results to 'num'
OFFSET num2      ← skip first 'num2' results.
```

3)
```
SELECT *
FROM table
ORDER BY column1 ASC, column2 DESC;
```
first condition is excuted first given most priority then second column / second condition.

4) HAVING CLAUSE - This clause used in association with group by clause applied to each group of results and similar to where clause just diff is its used with group by only.

```
SELECT column
FROM table
GROUP BY column
HAVING condition.
```
} hence used for applying condition with group by clause.

Q what is Group-By Clause? How is it used?
Q. what is the difference between Having & Where.

Page
Date / /

4) GROUPBY - Used to Group the column values

```
SELECT column
FROM table
GROUP BY column.
```

5) USING clause - It is used for joing it can be used instead of 'ON' in JOIN.

```
SELECT columnname
FROM tablename
JOIN tablename
USING (Column name)
```

Q. How to add record into table?

Camlin | Page
Date / /

INSERT Statement

INSERT - Used to insert new data into table

1) INSERT INTO SELECT Statement

```
INSERT INTO 'table name class'
  (Id, name, marks)
SELECT Id, name, marks
FROM table name class
WHERE (condition);
```

2) INSERT INTO Statement

i) Without specifying columns.

~~SELECT~~

```
INSERT INTO tablename class
VALUES (value1, value2, value3);
```

ii) With specifying columns to fill

~~St~~

```
INSERT INTO tablename class
(column1, column2, column3)
VALUES (value1, value2, value3);
```

* Take care of sequence and ~~amount~~ number of values passed and their column matches.
* we can use multiple 'Insert into' - value - statements at a time.
* we can insert default values also as ⇒

```
INSERT INTO tablename
  DEFAULT VALUES
```

It put NULL values.

UPDATE Statement

Camlin | Page
Date / /

UPDATE - Used to modify data that already exists. to retrieve the data to modify we use WHERE clause.

Syntax

```
UPDATE    tablename
SET       columnname = expression
WHERE     (Condition)
```

e.g.

```
UPDATE    tablename class
SET       name = 'Shubham'
          lastname = 'Sawant'
WHERE     id = 17133 ;
```

id	name	Lastname
17133	Shubham	Sawant

* We can set value to null by setting 'NULL'

DELETE Statement

DELETE - Used to remove one or more records from table. or whole tables. Data to delete retrieved by 'where' clause.

Syntax

i) Deleting the whole table

```
DELETE FROM tablename;
```

ii) Deliting Specific row

```
DELETE FROM table name
WHERE (conditions);
```

semicolon play imp. role.

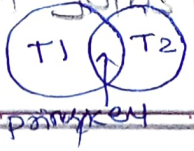
e.g.

```
DELETE FROM table name class
WHERE id = 17133;
```

it will delete row of shubham sawant.

Q What are the different joins in SQL?

JOIN



Camlin | Page
Date / /

1) OUTER JOIN - two tables joined at key columns but values where not matching either left or right sided values shown.

2) INNIR JOIN - two tables combined where key element of key columns matches.

People	
name	state
Marcus	CA
Jennifer	VA
Devin	MA
Britt	CA

states	
abbr	full
CA	California
DE	Delaware
VA	Virginia

Inner join

```
SELECT *
FROM People
JOIN States
ON people.state = States.abri
```

name	state	abr	full
Marcus	CA	CA	Califor
Jeniffer	VA	VA	virgin
Britt	CA	CA	Califo

where values matches.

Outer join

LEFT JOIN

```
SELET *
FROM People
LEFT JOIN States
ON People.State = States.abbr;
```

name	State	abbr	full
marcus	CA	CA	califor
Jennifer	VA	VA	Virgin
Devin	MA	null	null
Britt	CA	CA	Calu

RIGHT JOIN

```
SELECT *
FROM People
RIGHT JOIN
ON People.State = States.abbr;
```

name	state	abbr	full
marcus	CA	CA	califer
Jenifer	VA	VA	Virn
Brit	CA	CA	Calu
null	null	DE	Delun

Q. what is cartesian product of table (CrossJoin)

Camlin | Page
Date / /

FULL OUTER JOIN

```
SELECT *
FROM table1
FULL OUTER JOIN table2
ON table1.col1 = table2.col1
```

CROSS JOIN – when each row of first table combine with each row of second table. its a cartesian of cross Join. e.g if two ~~columns~~ tables 15 & 20 column cross join product will be 15x20=300 rows.

Syntax

```
SELECT *
FROM table1 CROSS JOIN table2
```

table1

id	name
1	Shubham
2	Rohit

table 2

id	surname
1	Sawant
2	Shinde

output

name	surnam
Shubham	Sawan
Shubham	Shind
Rohit	sawa
Rohit	Shi

It multiply all the results. Cross Product.

String

Camlin | Page
Date / /

Standerd SQL

```
SELECT 'a literal SQL string';
```

MySQL

```
SELECT "a literal SQL string";
```

Functions of String

1) LENGTH

```
SELECT name, LENGTH(name) AS Length
FROM country
```

(New column with length of strings)

name	Length
pune	4
satara	6

2) SUBSTR

```
SELECT released,
       SUBSTR(reased, 1, 4) AS Year,
       SUBSTR(realed, 6, 2) AS Month,
       SUBSTR(realed, 9, 2) AS Day,
FROM album
```

(column string; starting position character; No. of characters)

released	Year	Month	Day
1959-08-07	1959	08	07
1969-01-06	1969	01	06

3) TRIM

```
SELECT TRIM('string');      ⇒ string
SELECT LTRIM('  string');   ⇒ string
SELECT RTRIM('string  ');   ⇒ string
SELECT TRIM('...string---', '.-') ⇒ string
```

(string; what to trim from string)

4) UPPER , LOWER

i) SELECT LOWER(name)
FROM city
ORDER BY name;

name
pune
Satara

ii) SELECT UPPER(name)
FROM city
ORDER BY name

name
PUNE
SATARA

5) REPLACE

SELECT REPLACE(nam), 'a', '_')
FROM city

(Case sensitive)

name
Pune
S_t_r_

Numbers

Data types are system dependent. different on every system.

1) Integer
2) Real / float

(.1 + .2) * 10 = 1.0 + 2.0 ⇒ False

Sacrificing accuracy for scale so value are large but not precise in real numbers.

Functions

1) TYPE OF
i) SELECT ⇒ integer
TYPE OF (1+1);

ii) SELECT TYPEOF(1+1.0) ⇒ Real
iii) SELECT TYPEOF('Panda') ⇒ text
iv) SELECT TYPEOF('Panda' + 'Koala') ⇒ integer

↑
It gives text as result if system support concatenation.

2) Integer division

i) SELECT 1/2, 1.0/2; ⇒

1/2	1.0/2
0	0.5

ii) SELECT 17/5, 17.1.5; ⇒

17/5	17.1.5
3	2

iii) SELECT 2>3, 3<5; ⇒

2>3	3<5
0	1

Camlin Page
Date / /

3) CAST

```
SELECT CAST(1 AS REAL)/2; ⇒
```

CAST(1 AS REAL)/2
0.5

4) ROUND

```
SELECT ROUND(2.5555, 3); ⇒
```

ROUND(2.5555,3)
2.555

```
SELECT ROUND(2.5555, 0); ⇒
```

ROUND(2.5555,0)
3

Dates and times

Camlin | Page
Date / /

format

2018 - 03 - 08 15 : 32 : 47

← most significant less significant →

UTC - universal time coordinate, to avoid conflicts use standerd time

functions

1) SELECT DATETIME ('now'); ⇒
2) SELECT DATE ('now'); ⇒
3) SELECT TIME ('now'); ⇒
4) SELECT DATETIME ('now', '+3 hours', '+27 minutes', '-1 day', '+3 years');

costume function for complex ~~date time modification~~.
Note the plural and singulars also.

Outputs

1) DATETIME('now')	2) DATE('now')	3) TIME('now')
2019-10-29 21:21:32	2019-10-29	21:21:32

4) DATETIME ('now', '+3 hours', '+27 minutes', '-1 day', '+3 years')
2022-10-29 00:50:06

Q. what is groupby clause
Q. what are some aggregate functions

Aggregates

Camlin | Page | Date / /

Clause - GROUP BY - It is used with select statement to group the results of Query matches the values in columns and group them accordingly.

```
SELECT  column name (s)
FROM    table name
GROUPBY column name
```

functions

1) COUNT

```
SELECT  COUNT(*)
FROM    country;
```

COUNT(*)
10

Count of no. of rows in 'population' columns

2)

```
SELECT  COUNT(Population)
FROM    country;
```

COUNT(Population)
9

No. of rows from population column

2) MIN, MAX, SUM, AVG

```
SELECT  region, MIN(Population),
                MAX(Population),
                SUM(Population),
                AVG(Population)
FROM    country
GROUP BY region;
```

min of population
max of population
sum of all value of population
Average of value in population
grouping population in region.

outputs

region	MIN(Population)	MAX(Population)	SUM(Pop)	AVG(P
Africa	600	2000	10000	1000
America	700	4000	20000	1200
Asia	500	1000	40000	1500

Q. what is distinct ? How we use it

Camlin Page
Date / /

DISTINCT

```
SELECT COUNT (DISTINCT Population)
FROM Country
```

count (DISTINCT Population)
179

gives count of distinct values Ro only from Population column.

3) LAST(), FIRST - Returns last or first values.

concepts @ what is SQL
Subselect @ what are types of SQL statements

SQL - structured query language that lets us communicate with database to store, manipulate, retrieve data & lets you modify structure of database.

Database - considered as container of tables

table - Grids with rows and columns that holds data.

Query - single statement in SQL, SQL is case Insensitive but by conventions we use capital letters for query.

SQL commands

DDL	DML	DCL	TCL
Data definition language	Data manipulation language	Data control language	Transaction control language
commands like creating, deleting modyfying the databases, schema tables, rows, columns etc.	commands like create, manipulate the actual data	commands like Grant, Revoke	commands like commit, savepoint, rollback.
i) CREATE ii) ALTER iii) RENAME iv) DROP etc v) COMMENT vi) TRUNCATE	i) CRUD commands ii) WHERE clause	i) GRANT ii) REVOKE	i) COMMIT ii) SAVEPOI iii) ROLLBACK iv) SET TRANS

Q. why do we use constraints?
Q what the different constraints? Explain all
Q what are the types of keys? Explain all

Constraints - constraints limits the type of data be passed into database they can be appl to columns or whole table. they improve reliability of Data.

Common constraints

1) UNIQUE - No duplicate in given column
2) NOT NULL - No Null value
3) DEFUALT - If value not passed default val
4) CHECK - Ensures value in column satisfy give
5) INDEX - Retrive data from database quicke
6) PRIMERY KEY - combination of not null & unique
7) SECONDARY KEY - uniquely identify row recor from another table.

Keys - columns used to create relationships within ta and maintain uniqueness and retrive data faster.

Primary key - not null and unique values of column mo suitable to select as primary key with w we can join the tables. eg. id, Passportn

unique key - Similar to primary but allows one null in column

Alternate key - Candidate key not selected as primary ke

Super key - set of two or more columns that uniquel identify the rows e.g. combination of t candite keys, Primary key & othe key e

Foreign key - In Relationship with two tables primar of one table can be called foreign key f another table. it can accept null value null value allowed.

Candidate key - Any key suitable for selecting as pri key is candidate key.

Q. what is view?
Q. does view contain data? ⇒ No

Camlin | Page
Date / /

View

view - i) It's a virtual table not physically exists and its a result set of queries on the data.
ii) SQL view can be created by joining one or more tables.
iii) view don't have data and they can have Based on ~~another view~~.

Syntax

i) to create

```
CRAETE VIEW viewname AS
SELECT columnsname
FROM tablename
WHERE (conditions)
```

ii) to delete

```
DROP VIEW View name
```

iii) UPDATE VIEW - create and replace.

```
CREATE or REPLACE VIEW viewname AS
SELECT column name
FROM table name
WHERE (conditions)
```

DBMS
Database

Camlin | Page
Date / /

Data - Collection of small units of information like texts, pictures, media, numbers etc stored in pieces of paper or memory devices.

Database - i) Organised collection of data so it can be easily accessed and managed

ii) purpose of database is to operate a large ammount of information by storing, retrieving and manging data.

iii) Database handler create database in such way that only one set of software program provides access of data to all users.

iv) e.g. databases available like MySQL, Sybase, Oracle, SQL server etc.

Evolution of Databases

- 1968 – File Based
- 68-1980 – Hierarchial data model - IBM's IMS DBMS
  Parent child model
- 1971 – Network dat model - IDS (Integrated Datastore)
- 70-Present – Relational Database

main term.

Instance

Instance is table with row and column

schema

schema is the structure of instance like name of column, it's type, relationships.

Q what is DBMS & RDBMS

Camlin Page
Date / /

Cloud database

Give access to retrieve, manage data, store data via Clo

Advantage - i) we can access from anywhere
ii) Lower cost
iii) Automated Processes like recovery, autos

~~Disadvantage~~ -

Examples - (AWS) Amazon web service, Micro. SQL se
oracle Database cloud service, Google spanne

	DBMS -	RDBMS
Define	DBMS is tool or Software used to store retrive data	
Relation	DBMS is not RDBMS essentially	RDBMS is DBMS
Data	DBMS store data as a file in hierarchial or navigational form	RDBMS store data as tab in the format header wh are column name & ro their values.
Data relation	there isn't a relation in data	theres is a relation in d
Accessibl	only single data element can accessed by single user	multiple data elements accessed b multiple users at a tim by SQL queries.
Space	very small quantity data stored and take huge space	Large quantity data c be stored, takes less s
ACID	Don't have any security regarding Data manipulation	Provided ACID property
Normali sation	No normalisation present	Normalisation present

Q what is table?
Q what is schema?

Redundancy (not useful stuf) present in DBMS	No Redundancy allowed because of key & indexes.
Examples are XML, File system	Example - MySQL, Oracle, SQL Server

RDBMS - Relational database management system.
It consists of -

1) Table - instance
2) Schema - Structure of instance i.e name of column, type, Relation
3) Tuples - rows in table which holds data.
4) Fields/ - columns or entities of records. which holds info.
5) Null value - Null value treated differently than zero or space contained by field its a absence of value
6) Data Integrity -
   i) Entity integrity = No duplicate rows in table
   ii) Domain integrity - valid entries for column by restricting type, format and range.
   iii) Referential integrity = Row cannot delete which is used by other records.
   iv) User defined integrity - Enforces some specific rule that are defined by user.

Properties of RDBMS

1) ~~no~~ sequence of column is insignificant.
2) Each column have unique name
3) Values should be atomic
4) Each row should be unique.

Q. what is table? & difference of DELETE & TRUNCATE & DROP

Camlin | Page
Date / /

Table

Table - Grid formed by columns and rows which holds data.

In DBMS - table is relation & row is tuple.

syntaxes

1) creating table

```
CREATE TABLE   tablename
(columnname1   Datatype, constraints,
 columnname2   Datatype  constraints, );
```

e.g.

```
CREATE table for class
( id   INT. PRIMARY KEY,
 name  varchar(225). NOT NULL, DEFAULT 'Shubham'
 surname  varchar(225) NOT NULL,
 marks    INT    UNIQUE );
```

this will create table with name 'table forclass' and columns id, name, surname, marks where id column is primary key hence values entered must be unique and distinct. name, surname should not be null.

2) Deleting table

i) DROP TABLE tablename; - throw error if table not exists.

ii) DROP TABLE IF EXISTS tablename; - Don't Throw error

iii) TRUNCATE TABLE tablename

iv) DROP TABLE tablename

constraints - 1) DEFAULT
2) PRIMIRY KEY
3) UNIQUE
4) NOT NULL

Delete	Truncate	Drop
Delete table but space is not cleared	Deletes tables and clears it's occupied space Truncated table cannt roll backed.	Deletes table a clears all its records. and all relationshi

3) Renaming the table - change the name with two types of syntaxes

i) ALTER TABLE oldtable
RENAME TO newtable ;

2) RENAME oldtable TO newtable

4) Copying table - If we want to copy table to another table ~~des~~ (that another table is destination).

SELECT *
INTO destinationtable name
FROM table to copy

5) Temporary table - two type i) Local temp - deletes on user disco from inst
tablename

CREATE TABLE
or ## tablename
(id int,
name varchar(250))

ii) Global temp - does not d
tablename unless all
withdraw

Q. How to add column to table?

6 Altering column properties

i) TO ADD columns in table

```
ALTER TABLE tablename
ADD ( columnname datatype,
      columnname datatype,
      columnname datatype);
```

e.g.
```
ALTER TABLE Classroom
      ADD ( address varchar(250),
            city    varchar(250));
```
It will add two columns in existing table.

ii) To modify column data type

```
ALTER TABLE tablename
MODIFY (columnname1, Datatype,
        columnname2 Datatype);
```

iii) To rename column name.

```
ALTER TABLE tablename
RENAME COLUMN columnname TO newcolname;
```

iv) To delete column from table

```
ALTER TABLE tablename
DROP COLUMN columnname;
```

common mistakes

1) Accidental smart qoutes (' ' ~)
2) Incorrect order of clauses
3) Not checking scope of destructive action
4) Typos and syntax errors
5) Single qoutes for TEXT fields.
6) using copy/paste from formatted texts. can cause

```
CREATE TABLE table name
( Age int ,
 CHECK ( Age >=18),
  ID   int          NOT NULL
  Name varchar(250) DEFAULT 'Shubham'
  Surname varchar(250) UNIQUE
  marks  int          PRIMERY KEY
  Roll no. int        FOREIGN KEY REFERENCES table
```

Q. what is meant by subquery.

SubQueries

Camlin Page
Date / /

subqueries - nested query or inner query embedded within the where clause.

* subsquery returns data that is used as condition for further restrict data to retrive.

* Subquery used with SELECT, UPDATE, INSERT, DELETE statements alongwith operators like, =, <, >, <=, >=, IN, BETWEEN, etc.

1) Select

```
SELECT *
FROM costumers
WHERE ID IN ( SELECT ID
              FROM costumer
              WHERE Salary >4500);
```

2) Insert

```
INSERT INTO costumer
SELECT *
FROM costumerstable
WHERE ID IN ( SELECT ID
              FROM coustemertable);
```

Inesert whole table (coustemerstable) into table (costumer)

Data returned by subquery inserted into another table

3) Update

```
UPDATE costumer
SET salary = salary * 2
WHERE age IN ( SELECT age
               FROM costumerstable
               WHERE age >=27);
```

Data returned is used as condition for updating data here age limit is got from 'costumerstable' and salary updated accordingly in 'costremer' hence operation eliminates the need for joining.

Q. How many comparison operators used

4) Delete

Similar to update

```
DELETE
FROM costumer
WHERE age IN (SELECT age
              FROM costumer table
              WHERE age >= 27);
```

Advantages - Subqueries are useful as they per-operations which would require complex join and unions. and are easy to read.

Disadvantages - It take longer to execute than join.

Comparison operators in SQL subquery

1) IN
2) ANY
3) ALL

Q what is normalisation? Types?

Normalisation

Camlin Page
Date / /

Normalisation - process of reducing Data redundancy and integrating data. to make easy to handle data & clear space.

Data Anomalies.

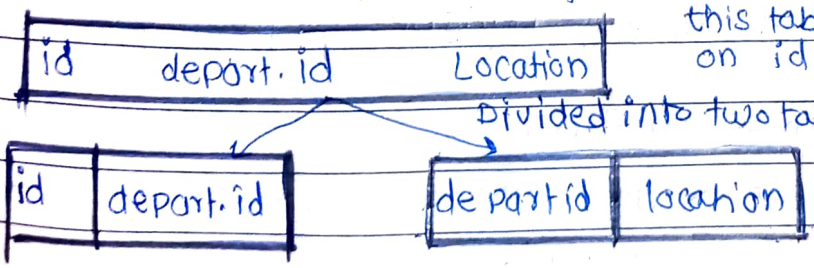
Insertion Updation Anomalies - If we put multiple values in table but leave a certain filed for all rows as Null its Insertion anomali

Updation Anomalies - If whole Record is to be updated but leaves certain data unupdated

Deletion Anomdjie - If two records are related to each other one of them to be deleted and we lost whole data.

1) 1st Normal form - Atomicity i.e. Single cell should not hold multiple values. e.g. two phone no. in phone no. column for single costumer we create two rows in that case.

2) 2nd normal form - 1st normal form should satisfied all non key attributes should be fully functional dependent on the primary key only.



this table have deport id depends on id and location depends on deport. id hence their is non key location is dependent on composite key not on primary key

Divided into two table

After dividing into two columns now every column has only one primary key and attributes depends on those keys only.

3) 3rd normal form - 2nd should satisfy.
No non-prime attributes should have transitive (par dependency (A→B, B→C then A→C). Shouldn't exists. an field only depends on primary key.

Std.id	std name	Address	subject id	subject

Here Std.id → subid → sub
hence ~~partial~~ transitive dependency.
to solve the prob we do divide it into two

Std id.	Std Name	address	Subject id

Now all attributes depends on std.id only

Subejest id	Sub

subject depe on subject

4) Boyce Codd BC Normal form - Should satisfy 3rd form. Similar to 3rd but more strict and every functional dependency is the super key.

EM.ID EM.County EMPEPT DEPT TYPE D

Here EM.ID → EM.County
EM DEPT → (DEPT TYPE , DEPT NO)
we divide it into three table

EM.ID	EM.COUNTY

EM DEPT	DEPT TYPE	DEPT NO

EM ID	EM DEPT

} fully functional dependency on primary
} super key

Q. what are transactions? and their controls?
controls like commit, Rollback, savepoint, set transaction

Transaction

Q. what is commit,

Transaction - It is a group of operations where any one of operation fails from group data base is restored to initial stage.

Syntax

```
BEGIN TRANSACTION          ← Starting Transaction
        INSERT INTO table-1
        INSERT INTO table-2     } Code to be executed as a single unit (Transaction)
        INSERT INTO table-3
END TRANSACTION            ← END of Transaction
```

Trasaction - It improves performance and liability. Speed of operations improved by transcation.

1) ROLLBACK - If any error occur in transaction we need to reverse the change then we use ROLLBACK.

```
DELETE FROM student
WHERE student id = 1
ROLLBACK;
```

} if its fails to do action we desire then we perform Rollback

2) COMMIT - If changes done successfully as we need to confirm we commit.

```
DELETE FROM student
WHERE student id=1;
COMMIT;
```

} successfull.

3) SAVEPOINT - Used to set point where ROLLBACK done

4) SET TRANSACTION - set name of transaction.

Q. What are the properties of transactions? (ACID)

ACID - Transcation must maintain ACID properties

Atomicity - Atomicity specifies transaction should be wheather it's done or fails it should be atomic unit.

Consistancy - Transcation should leave changes consistent should not lead to adverse effects on d

Durability - Database should be durable enough to data and all its updates even if system fails or starts once system fails then transaction should done when system re

Isolation - Each transaction should be treate as the only transaction it should not interfere with other transactions.

States of transactions