Basic

1. SELECT and WHERE for filtering and selection**-done**

2. COUNT, SUM, MAX, GROUP BY, HAVING for aggregating data" -

3. DISTINCT, COUNT DISTINCT for producing useful distinct lists and distinct aggregates -**DONE**

4. OUTER (e.g. LEFT) and INNER JOIN when/where to use them-DONE

5. Strings and time conversions-DONE

6. UNION and UNION ALL.-DONE

Intermediate:

1. DML/DDL/DCL concepts-DONE

2. Handling NULLs creatively (e.g. with COALESCE)-DONE

3. Subqueries and the impact of subqueries on efficiency of the queryPENDING

4. Temporary tables

5. Self joins DONE

6. Window functions like PARTITION, LEAD, LAG, NTILE

7. UDFs (user defined functions)

8. Use of in5dexes in querying to make operations faster.

What is Database?

Database is a collection of data stored in a format that can easily be accessed.

CREATE DATABASE:

Retrieve data from particular database / Select data from table

USE sql\_Store;

SELECT \*

From Customers

--WHERE customer\_id = 1;( -- used to make a comment)

SELECT CLAUSE

SELECT Last name,first\_name,points

points \* 10 +100 AS discount\_ factor. (AS alias)

FROM Customers;

SELECT DISTINCT state

FROM Customers;

WHERE CLAUSE

SELECT \*

From Customers;

WHERE birthdate>’1999-01-01’ OR (points>1000 AND state=”VA”)

ORDER BY:

SELECT \*

From Customers

WHERE customer\_id = 1

ORDER BY firstname DESC ,state;

LIMIT CLAUSE:

SELECT \*

From Customers;

WHERE birthdate>’1999-01-01’ OR (points>1000 AND state=”VA”)

LIMIT BY 10;

GROUP-BY CLAUSE

all (\*) the columns should be mentioned in in GROUP BY clause SELECT clause mentioned.

SELECT DEPT\_ID ,SALARY

FROM DEPT

GROUP BY Dept no

MAX FUNCTION:

SELECT MAX(SALARY) AS HIGHEST SALARY,

FROM EMP

GROUP BY Dept no

SUM FUNCTION:

SELECT DEPT\_ID,SUM (SALARY) AS TOTAL\_SALARY,

FROM EMP

COUNT CLAUSE:

SELECT COUNT (emp\_ID) AS total\_emp

from DEPT

GROUP BY Dept no

COUNT & DISTINCT:

SELECT count(distinct customer\_id) as unique\_customer

FROM subscriptions

This statement will count the number of unique values in the specified column and calculate the average of all values in the column.

HAVING CLAUSE

SELECT DEPT NO,

AVG (SAL) AS AVGSAL,

MIN (SAL) AS MINSAL,

MAX(SAL) AS MAX SAL,

SUM (SAL) AS SUMSAL,

FROM EMPLOYEE

GROUP BY DEDT NO

HAVING (SAL) > 30000)

STRINGS AND TIME CONVERSIONS

-- Convert a string to a date

SELECT

CAST('2022-12-23' AS DATE) as date\_string,

-- Extract the month from a date

EXTRACT(MONTH FROM CAST('2022-12-23' AS DATE)) as month,

-- Convert a timestamp to a string

TO\_CHAR(CURRENT\_TIMESTAMP, 'DD-Mon-YYYY HH24:MI:SS') as current\_time

FROM dual;

UNION:

SELECT \* FROM EMP;

UNION

SELECT \* FROM SALARY;

UNION ALL :

SELECT \* FROM EMP

UNION ALL

SELECT \* FROM SALARY;

JOINS : **OUTER (e.g. LEFT) and INNER JOIN when/where to use them**

In general, you would use a LEFT JOIN when you want to include all rows from the first table, even if there is no match in the second table.

SELECT \* FROM EMP

LEFT JOIN SALARY

ON EMP.EMP\_ID = SALARY. EMP\_ID ;

You would use an INNER JOIN when you only want to include rows that have a match in both tables.

SELECT \* FROM EMP

INNER JOIN SALARY

ON EMP.EMP\_ID = SALARY. EMP\_ID

WHERE EMP.salary = 30000 ;

SELF JOINS:

SELECT e1.first\_name1, s2.last\_name

FROM EMP AS e1

JOIN SALARY AS s2 ON e1.emp\_id = s2.emp\_id

WHERE e1.salary = 30000;

**DML/DDL/DCL concepts**

**DML:**

SELECT \* FROM customers

WHERE city = 'San Francisco';

This statement uses the SELECT DML statement to retrieve all rows from the "customers" table where the city is "San Francisco".

INSERT INTO orders (customer\_id, product\_id, quantity) VALUES (1, 2, 5);

This statement uses the INSERT DML statement to insert a new row into the "orders" table with the specified values for the "customer\_id", "product\_id", and "quantity" columns.

UPDATE customers

SET email = 'newemail@example.com'

WHERE customer\_id = 3;

This statement uses the UPDATE DML statement to update the email address of the customer with an ID of 3 in the "customers" table.

DELETE FROM orders

WHERE order\_date < '2021-01-01';

This statement uses the DELETE DML statement to delete all rows from the "orders" table where the "order\_date" is before January 1, 2021.

**DCL :**

GRANT SELECT, INSERT ON table\_name TO user1;

This statement grants the specified permissions (in this case, SELECT and INSERT) to the specified user (user1) on the specified table.

REVOKE SELECT, INSERT ON table\_name FROM user1;

This statement revokes the specified permissions (in this case, SELECT and INSERT) from the specified user (user1) on the specified table.

**DDL:**

CREATE TABLE SALARY (column\_name1 datatype, column\_name2 datatype);

This statement creates a new table with the specified column names and data types.

ALTER TABLE table\_name

ADD column\_name3 datatype;

This statement adds a new column to the table with the specified name and data type.

DROP TABLE table\_name;

This statement deletes the specified table from the database.

Function

LTRIM (char-expr) => returns a character expr after it remo

TRIM (Chas\_expr)) returns a string after truncating all mo

CHAR (int, expr) CHARINDER (expr Jo find, => convert ⇒ Searches exp to search [, start, 10c]) return its an int ASCM code to a chara an exp for another expression rerwin its string position of found.

STR (expression)

returns character data converted from a CONCAT (SITI, SW2, S3...) => resins a string after concatenating 20