**DML commands**

**M ASWIN KISHORE, S5 CS2 41**

**Questions**:

1. Insert a single record in the following tables Customer\_db, item\_db, order\_db

insert into customer values(1, "MAK", "221B", "Baker Stret",904998900);

insert into item\_db values(1, 'Apple', 20, 47);

insert into order\_db values(12, 1, 1, '2021-12-03', '2021-12-04', '2021-12-03' ,"R");

1. Insert multiple records in the following tables

insert into item\_db values

(2, 'Mango', 60, 20),

(3, 'JackFruit', 40, 90);

1. Display the records in ascending order by delivery date in order\_db

SELECT \* FROM order\_db order by delivery\_date;

+----------+---------+-----------+------------+-------------+---------------+--------------+

| order\_id | cust\_id | item\_code | order\_date | expiry\_date | delivery\_date | payment\_mode |

+----------+---------+-----------+------------+-------------+---------------+--------------+

| 1 | 1 | 1 | 2021-12-03 | 2021-12-04 | 2021-12-03 | R |

| 2 | 2 | 3 | 2021-12-03 | 2021-12-23 | 2021-12-13 | C |

| 3 | 2 | 3 | 2021-12-03 | 2021-12-01 | 2021-12-13 | C |

+----------+---------+-----------+------------+-------------+---------------+--------------+

1. Display the records in descending order by unit price in item\_db

SELECT \* FROM item\_db order by delivery\_date;

+-----------+-----------+-------+------------+

| item\_code | item\_name | stock | unit\_price |

+-----------+-----------+-------+------------+

| 3 | JackFruit | 40 | 90 |

| 1 | Apple | 20 | 47 |

| 2 | Mango | 60 | 20 |

+-----------+-----------+-------+------------+

1. Display the records in order\_db where order\_id >1 and <7

SELECT \* FROM order\_db WHERE order\_id > 1 AND order\_id <7;

+----------+---------+-----------+------------+-------------+---------------+--------------+

| order\_id | cust\_id | item\_code | order\_date | expiry\_date | delivery\_date | payment\_mode |

+----------+---------+-----------+------------+-------------+---------------+--------------+

| 2 | 2 | 3 | 2021-12-03 | 2021-12-23 | 2021-12-13 | C |

| 3 | 2 | 3 | 2021-12-03 | 2021-12-01 | 2021-12-13 | C |

+----------+---------+-----------+------------+-------------+---------------+--------------+

1. Delete the order from order\_db where expiry date < current date

DELETE FROM order\_db WHERE expiry\_date < current\_date();

1. Duplicate the table of item\_db as test\_db

CREATE TABLE test\_db as SELECT \* FROM item\_db;

1. Select customer name from customer\_db whose name starts with alphabet A.

SELECT cust\_name FROM customer WHERE cust\_name LIKE "A%";

+-----------+

| cust\_name |

+-----------+

| Amal |

+-----------+

1. Display order\_id and cust\_id from order\_db

SELECT order\_id,cust\_id from order\_db;

+----------+---------+

| order\_id | cust\_id |

+----------+---------+

| 2 | 2 |

| 3 | 2 |

| 1 | 1 |

+----------+---------+

1. Truncate the table item\_db

TRUNCATE TABLE order\_db;

1. Display order\_id , customer\_id from order\_db whose month of delivery is current month.

select order\_id,cust\_id from order\_db where delivery\_date BETWEEN '2021-12-01' and '2021-12-31';

+----------+---------+

| order\_id | cust\_id |

+----------+---------+

| 2 | 2 |

| 3 | 2 |

| 1 | 1 |

+----------+---------+

1. Find the average of order quantity with item code =256.

SELECT AVG(order\_qty) FROM order\_db WHERE item\_code=256;

+----------------+

| AVG(order\_qty) |

+----------------+

| 20.0000 |

+----------------+

1. What is the item with the highest unit price?

SELECT \* FROM item\_db WHERE unit\_price = (SELECT MAX(unit\_price) FROM item\_db);

+-----------+-----------+-------+------------+

| item\_code | item\_name | stock | unit\_price |

+-----------+-----------+-------+------------+

| 3 | JackFruit | 40 | 90 |

+-----------+-----------+-------+------------+

1. What is the cheapest item?

SELECT \* FROM item\_db WHERE unit\_price = (SELECT MIN(unit\_price) FROM item\_db);

+-----------+-----------+-------+------------+

| item\_code | item\_name | stock | unit\_price |

+-----------+-----------+-------+------------+

| 2 | Mango | 60 | 20 |

| 4 | Spota | 200 | 20 |

+-----------+-----------+-------+------------+

1. How many orders were made for item with itemcode 256

SELECT COUNT(\*) FROM order\_db WHERE item\_code=256;

+----------+

| COUNT(\*) |

+----------+

| 1 |

+----------+

1. How many items have unit price between 100 and 200.

SELECT COUNT(\*) FROM item\_db WHERE unit\_price BETWEEN 100 AND 200

+----------+

| COUNT(\*) |

+----------+

| 0 |

+----------+

1. What is the average unit price?

SELECT AVG(unit\_price) FROM item\_db;

+-----------------+

| AVG(unit\_price) |

+-----------------+

| 44.2500 |

+-----------------+

1. Display the orderid and delivery date with heading order code and date of delivery.

select order\_id AS "order code", delivery\_date AS "date of delivery" FROM order\_db;

+------------+----------------+

| order code | date of delivery |

+------------+----------------+

| 2 | 2021-12-13 |

| 3 | 2021-12-13 |

| 1 | 2021-12-03 |

| 4 | 2021-12-13 |

+------------+----------------+

1. Display the name of customers which contain occurances of a and j in the same name.

SELECT cust\_name FROM customer WHERE

cust\_name IN (SELECT cust\_name FROM customer WHERE cust\_name LIKE "%a%")

AND

cust\_name IN (SELECT cust\_name FROM customer WHERE cust\_name LIKE "%j%");

+-----------+

| cust\_name |

+-----------+

| Ajmal |

+-----------+

1. What is the length of shortest name?

SELECT MIN(LENGTH(cust\_name)) FROM customer

+------------------------+

| MIN(LENGTH(cust\_name)) |

+------------------------+

| 3 |

+------------------------+

1. Create table deliverdb with same structure as orderdb.

CREATE TABLE deliver\_db as (SELECT \* FROM order\_db where cust\_id is NULL);

1. Display the records of tables orderdb deliverydb using union operator

SELECT \* from deliver\_db UNION select \* from order\_db;

+----------+---------+-----------+------------+-------------+---------------+--------------+-----------+

| order\_id | cust\_id | item\_code | order\_date | expiry\_date | delivery\_date | payment\_mode | order\_qty |

+----------+---------+-----------+------------+-------------+---------------+--------------+-----------+

| 2 | 2 | 3 | 2021-12-03 | 2021-12-23 | 2021-12-13 | C | 50 |

| 3 | 2 | 3 | 2021-12-03 | 2021-12-01 | 2021-12-13 | C | 10 |

| 1 | 1 | 1 | 2021-12-03 | 2021-12-04 | 2021-12-03 | R | 20 |

| 4 | 1 | 256 | 2021-12-02 | 2021-12-01 | 2021-12-13 | R | 20 |

+----------+---------+-----------+------------+-------------+---------------+--------------+-----------+

1. Display the records having order id common for both tables orderdb and deliverydb using intersect operator

SELECT e.order\_id,e.cust\_id,e.item\_code,e.order\_date,e.expiry\_date,e.delivery\_date from deliver\_db as e

INNER JOIN

order\_db

ON e.order\_id = order\_db.order\_id;

+----------+---------+-----------+------------+-------------+---------------+

| order\_id | cust\_id | item\_code | order\_date | expiry\_date | delivery\_date |

+----------+---------+-----------+------------+-------------+---------------+

| 4 | 1 | 256 | 2021-12-02 | 2021-12-01 | 2021-12-13 |

+----------+---------+-----------+------------+-------------+---------------+

1. Display the orderid of the order that is not delivered yet.

SELECT order\_id FROM order\_db WHERE delivery\_date > '2021-12-03';

+----------+

| order\_id |

+----------+

| 2 |

| 3 |

| 4 |

+----------+