By SDRA AWAMEH 202210368

DBP ASSINMENT 1

Agenda

1. Q1
2. Q2
   - .Q2.A
   - .Q2.B
   - .Q2.C
3. Q3
   - .Q3.A
   - .Q3.B

Q1

CODE

```
-- Create employee table
CREATE TABLE employee (
 ID INT PRIMARY KEY,
 person_name VARCHAR(100),
 street VARCHAR(100),
 city VARCHAR(100));

-- Create company table
CREATE TABLE company (
 company_name VARCHAR(100) PRIMARY KEY,
 city VARCHAR(100)
);
```

Q1

CODE

```
-- Create works table
CREATE TABLE works (
  ID INT PRIMARY KEY,
  company_name VARCHAR(100),
  salary DECIMAL(15, 2),
  FOREIGN KEY (ID) REFERENCES employee(ID) ON DELETE CASCADE,
FOREIG KEY(company_name)REFERENCES company(company_name) ON DELETE CASCADE Referencing
company name
);

-- Create managers table
CREATE TABLE managers (
   ID INT PRIMARY KEY,
  FOREIGN KEY (ID) REFERENCES employee(ID) ON DELETE CASCADE,
 FOREIGN KEY (manager_id) REFERENCES employee(ID) ON DELETE SET NULL );
```

Q1

```
CREATE TABLE managers (
ID INT PRIMARY KEY,
manager_id INT,
FOREIGN KEY (ID) REFERENCES employee(ID) ON DELETE CASCADE,
FOREIGN KEY (manager_id) REFERENCES employee(ID) ON DELETE SET NULL
);
```

Results Explain Describe Saved SQL History

ID	MANAGER_ID
2	1
3	1
4	2
5	2

```
CREATE TABLE employee (
ID INT PRIMARY KEY,
person_name VARCHAR(100) NOT NULL,
street VARCHAR(100),
city VARCHAR(100)
);
```

Results Explain Describe Saved SQL History

ID	PERSON_NAME	STREET	CITY
1	Alice	123 Main St	New York
2	sdra	456 Elm St	Los Angeles
3	rasha	789 Oak St	Chicago
4	mohamad	321 Pine St	Houston
5	leen	654 Maple St	Phoenix

Autocommit Rows 10

```
CREATE TABLE company (
company_name VARCHAR(100) PRIMARY KEY,
city VARCHAR(100)  );
```

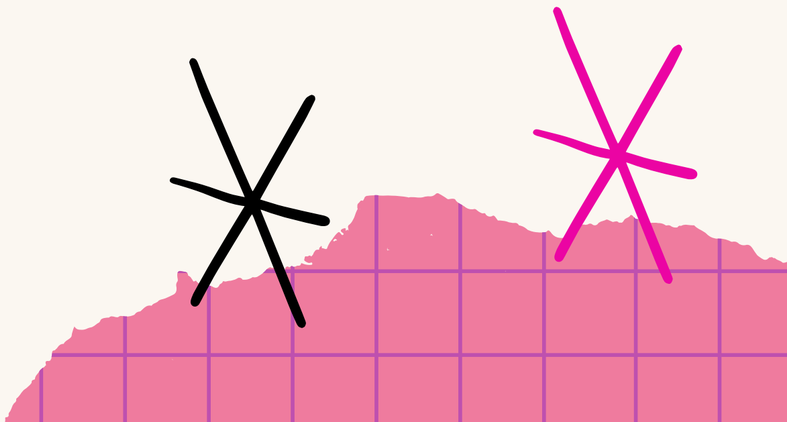
Results Explain Describe Saved SQL H

COMPANY_NAME	CITY
TechCorp	San Francisco
InnovateInc	New York
Alpha Solutions	Los Angeles

```
CREATE TABLE works (
ID INT PRIMARY KEY,
company_name VARCHAR(100) NOT NULL,
salary DECIMAL(10, 2),
FOREIGN KEY (ID) REFERENCES employee(ID) ON DEL
);
```

Results Explain Describe Saved SQL History

ID	COMPANY_NAME	SALARY
1	TechCorp	90000
2	HealthInc	85000
3	TechCorp	78000
4	FinServe	95000
5	HealthInc	82000



Q2.A

A Find the ID of each customer of the bank who has an account but not a loan.?

Answer:

```
SELECT dep.id
FROM depositor dep
WHERE NOT EXISTS (
    SELECT bor.id
    FROM borrower bor
    WHERE bor.id = dep.id
);
```

```
SELECT dep.id
FROM depositor dep
WHERE NOT EXISTS (
    SELECT bor.id
    FROM borrower bor
    WHERE bor.id = dep.id
);
```

Results Explain Describe Sa

ID
12345
23456
45678

Q2.B

B Find the ID of each customer who lives on the same street and in the same city as customer '12345'.?

Answer:

```
SELECT customer_id
FROM customer
WHERE (customer_street,
customer_city) =
(
   SELECT customer_street,
customer_city
   FROM customer
   WHERE customer_id = 12345
)
AND customer_id != 12345;
```

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CITY
12345	John Doe	Main St	Harrison
23456	Jane Smith	Main St	Harrison
34567	Alice Johnson	Broadway	New York
45678	Bob Brown	Main St	Harrison
56789	Eve White	5th Ave	Chicago

```
SELECT customer_id
FROM customer
WHERE (customer_street, customer_city) =
(
    SELECT customer_street, customer_city
    FROM customer
    WHERE customer_id = 12345
)
AND customer_id != 12345;
```

Results Explain Describe Saved SQL Histo

CUSTOMER_ID
23456
45678

Q2.C

C Find the name of each branch that has at least one customer who has an account in the bank and who lives in “Harrison”.?

Answer:

```
SELECT DISTINCT acc.branch_name
 FROM account acc
 JOIN depositor dep ON
acc.account_number =
dep.account_number
 JOIN customer cust ON dep.id =
cust.customer_id
 WHERE cust.customer_city = 'Harrison';
```

```
SELECT DISTINCT acc.branch_name
FROM account acc
JOIN depositor dep ON acc.account_number = dep.account_number
JOIN customer cust ON dep.id = cust.customer_id
WHERE cust.customer_city = 'Harrison';
```

Results | Explain | Describe | Saved SQL | History

BRANCH_NAME
Harrison Branch

Q3.A

Question:

A From the demand table, find the cumulative total sum for qty?

```
SELECT day, qty,
   SUM(qty) OVER (ORDER BY day
ROWS BETWEEN UNBOUNDED
PRECEDING AND CURRENT ROW) AS
cumulative_sum
FROM
   demand;
```

Answer:

```
SELECT
    day,
    qty,
    SUM(qty) OVER (ORDER BY day ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cumulative_sum
FROM
    demand;
```

Results | Explain | Describe | Saved SQL | History

DAY	QTY	CUMULATIVE_SUM
1	10	10
2	6	16
3	21	37
4	9	46
6	12	58
7	18	76
8	3	79
9	6	85
10	23	108

Q3.B

Question:

B From the demand table, find the cumulative total sum for qty?

```
SELECT  product ,day, qty, rank
FROM  (
     SELECT  product, day,qty,
      RANK() OVER (PARTITION BY
product ORDER BY qty ASC) AS
rank
     FROM demand2
    )
WHERE
  rank <= 2;
```

Answer:

```
SELECT    product ,day, qty, rank
FROM   (
         SELECT  product, day,qty,
          RANK() OVER (PARTITION BY product ORDER BY qty ASC) AS rank
         FROM demand2
       )
WHERE
     rank <= 2;
```

Results Explain Describe Saved SQL History

PRODUCT	DAY	QTY	RANK
A	2	6	1
A	4	9	2
B	3	3	1
B	4	6	2

4 rows returned in 0.00 seconds Download

THANK YOU!

SDRA AWAMEH 202210368