

ASSIGNMENT - RDBMS and SQL

1. Write a **SELECT** query to retrieve all columns from a 'customers' table, and modify it to return only the customer name and email address for customers in a specific city.

```
CREATE TABLE customers (Cust_Id number(3) Primary key, Cust_Name varchar(15) not null, Cust_Email varchar(20) not null, Cust_City varchar(20));
```

```
insert into customers values(1, 'Resh', 'resh@wipro.com', 'Jabalpur');
```

```
insert into customers values(2, 'Jignesh', 'jignesh@wipro.com', 'Jhasi');
```

```
insert into customers values(3, 'Mohan', 'mohan@wipro.com', 'Jabalpur');
```

```
desc customers;
```

```
Select * from customers;
```

CUST_ID	CUST_NAME	CUST_EMAIL	CUST_CITY
1	Resh	resh@wipro.com	Jabalpur
2	Jignesh	jignesh@wipro.com	Jhasi
3	Mohan	mohan@wipro.com	Jabalpur

3 rows returned in 0.01 seconds

```
Select Cust_Name, Cust_Email From customers where Cust_City = 'Jabalpur';
```

CUST_NAME	CUST_EMAIL
Resh	resh@wipro.com
Mohan	mohan@wipro.com

2 rows returned in 0.00 seconds

2. Craft a query using an **INNER JOIN** to combine 'orders' and 'customers' tables for customers in a specified region, and a **LEFT JOIN** to display all customers including those without orders.

```
SELECT customers.Cust_Name, orders.OdrID
```

```
FROM customers
```

```
INNER JOIN orders
```

```
ON customers.Cust_ID = orders.OdrId
```

```
WHERE Cust_City = 'Jabalpur'
```



ASSIGNMENT - RDBMS and SQL

CUST_NAME	ODRID
Resh	1
Mohan	3

2 rows returned in 0.01 seconds

```
SELECT customers.Cust_Name, orders.OdrID
FROM customers
LEFT JOIN orders
ON customers.Cust_ID = orders.OdrID
```

CUST_NAME	ODRID
Resh	1
Jignesh	2
Mohan	3
Rohan	-

4 rows returned in 0.00 seconds

3. Utilize a subquery to find customers who have placed orders above the average order value, and write a UNION query to combine two SELECT statements with the same number of columns.

The SQL subquery to find customers who have placed orders above the average order value is as follows:

Select * from customers;

CUST_ID	CUST_NAME	CUST_EMAIL	CUST_CITY
1	Resh	resh@wipro.com	Jabalpur
2	Jignesh	jignesh@wipro.com	Jhasi
3	Mohan	mohan@wipro.com	Jabalpur

3 rows returned in 0.01 seconds

Select * from orders;

ODRID	ODRNAME	ODRPRICE
1	Bag	300
3	Books	500
2	Pencil	10

3 rows returned in 0.00 seconds

ASSIGNMENT - RDBMS and SQL

Select Cust_Name From customers

Where Cust_Id In (Select OdrId From orders

Where OdrPrice > (select Avg(OdrPrice) From orders)) ;

CUST_NAME	CUST_EMAIL
Resh	resh@wipro.com
Mohan	mohan@wipro.com

2 rows returned in 0.00 seconds

The SQL UNION query to combine two SELECT statements with the same number of columns is as follows:

Select Cust_Id, Cust_Name

From customers

UNION

Select OdrId, OdrName

From orders;

CUST_ID	CUST_NAME
1	Bag
1	Resh
2	Jignesh
2	Pencil
3	Books
3	Mohan
4	Rohan

7 rows returned in 0.00 seconds

4. Compose SQL statements to BEGIN a transaction, INSERT a new record into the 'orders' table, COMMIT the transaction, then UPDATE the 'products' table, and ROLLBACK the transaction.

CREATE TABLE ORDERS(odrId number(3) PRIMARY KEY, odrPrice number(7,2) not null);
desc orders;



ASSIGNMENT - RDBMS and SQL

```
CREATE TABLE products ( PId number(3) PRIMARY KEY, PName varchar(15) not null,  
PQuantity number(5) not null, odrId number(3), FOREIGN KEY (odrId) REFERENCES  
Orders(odrId));  
decs products;  
Insert into orders values(1, 300.0);  
Insert into orders values(2, 10.0);  
Insert into products values(101, ' Bag ', 3, 1);  
Insert into products values(102, ' Pencil ', 10, 2);  
Select * from orders;
```

PID	PNAME	PQUANTITY	ODRID
101	bag	10	1
102	Pencil	15	2

2 rows returned in 0.00 seconds

Commit;

Update Products Set PQuantity = 5

Where PId = 1;

Rollback;

Select * from products;

PID	PNAME	PQUANTITY	ODRID
101	bag	10	1
102	Pencil	15	2

2 rows returned in 0.00 seconds

5. Begin a transaction, perform a series of INSERTs into 'orders', setting a SAVEPOINT after each, rollback to the second SAVEPOINT, and COMMIT the overall transaction.

set autocommit=false;

create table orders(odrId number(3) PRIMARY KEY , odrName varchar(15) not null, odrPrice number(7,2) not null);

desc orders;

Insert into orders values(1,'Bag',300.0);

Insert into orders values(2,'Pencil',10.0);



ASSIGNMENT - RDBMS and SQL

Insert into orders values(3,'Books',500.0);

Insert into orders values(4,'Laptop',30000.55);

Select * from orders;

ODRID	ODRNAME	ODRPRICE
1	Bag	300
2	Pencil	10
3	Books	500
4	Laptop	30000.55

4 rows returned in 0.00 seconds

delete from orders where odrId = 2;

Select * from orders;

ODRID	ODRNAME	ODRPRICE
1	Bag	300
3	Books	500
4	Laptop	30000.55

3 rows returned in 0.01 seconds

savepoint a;

delete from orders where odrId = 3;

Select * from orders;

ODRID	ODRNAME	ODRPRICE
1	Bag	300
4	Laptop	30000.55

2 rows returned in 0.00 seconds

savepoint b;

delete from orders where odrId = 4;

Select * from orders;

ODRID	ODRNAME	ODRPRICE
1	Bag	300

1 rows returned in 0.00 seconds

commit;

rollback;

ASSIGNMENT - RDBMS and SQL

Select * from orders;

ODRID	ODRNAME	ODRPRICE
1	Bag	300
3	Books	500
2	Pencil	10

3 rows returned in 0.00 seconds

6. Draft a brief report on the use of transaction logs for data recovery and create a hypothetical scenario where a transaction log is instrumental in data recovery after an unexpected shutdown.

Transaction logs play a crucial role in data recovery in database management systems (DBMS). They record all the actions performed by transactions providing a reliable and consistent way to recover data in the event of unexpected shutdowns, hardware failures, or other system issues. Transaction logs enable the system to have robust recovery and data tracking capabilities, ensuring the durability and recoverability of the database.

Hypothetical Scenario: Transaction Log Instrumental in Data Recovery

Let's consider a hypothetical scenario where a transaction log is instrumental in data recovery after an unexpected shutdown.

Scenario:

A company, Wipro, operates a customer relationship management (CRM) system to manage customer data. The CRM system is critical for the company's operations, and any data loss or corruption could have severe consequences.

Event:

One day, there is an unexpected power outage that causes the CRM system to shut down abruptly. When the system is restarted, it fails to load properly, and some customer data appears to be missing or corrupted.

Data Recovery Process:

To recover the lost or corrupted data, the system administrator initiates the data recovery process, leveraging the transaction log.

- 1. Identifying the Last Consistent State:** The system administrator examines the transaction log to identify the last consistent state of the database before the unexpected

ASSIGNMENT - RDBMS and SQL

shutdown. The transaction log contains a record of all the transactions that have been performed on the database, allowing the administrator to determine the point at which the data was in a consistent state .

2. **Undoing Changes with Log Records:** Using the log records stored in the transaction log, the system administrator begins the recovery process by undoing the changes made by the failed transactions. The log records provide a detailed account of the modifications made, allowing the administrator to reverse the effects of the incomplete or failed transactions and restore the database to its previous state .
3. **Redoing Changes with Log Records:** After undoing the changes, the system administrator uses the log records stored in the transaction log to redo the changes made by the transactions that were in progress at the time of the unexpected shutdown. By reapplying the changes recorded in the log, the administrator ensures that the database reflects the modifications made by the transactions that were interrupted .
4. **Verifying Data Consistency:** Throughout the recovery process, the system verifies the consistency and validity of the log data. Checksums are used to ensure that the log data is consistent and has not been corrupted. If any invalid data is detected, appropriate actions are taken to prevent its use and ensure data integrity .

By following these steps and leveraging the transaction log, the system administrator successfully recovers the lost or corrupted data in the CRM system, restoring it to a consistent state.