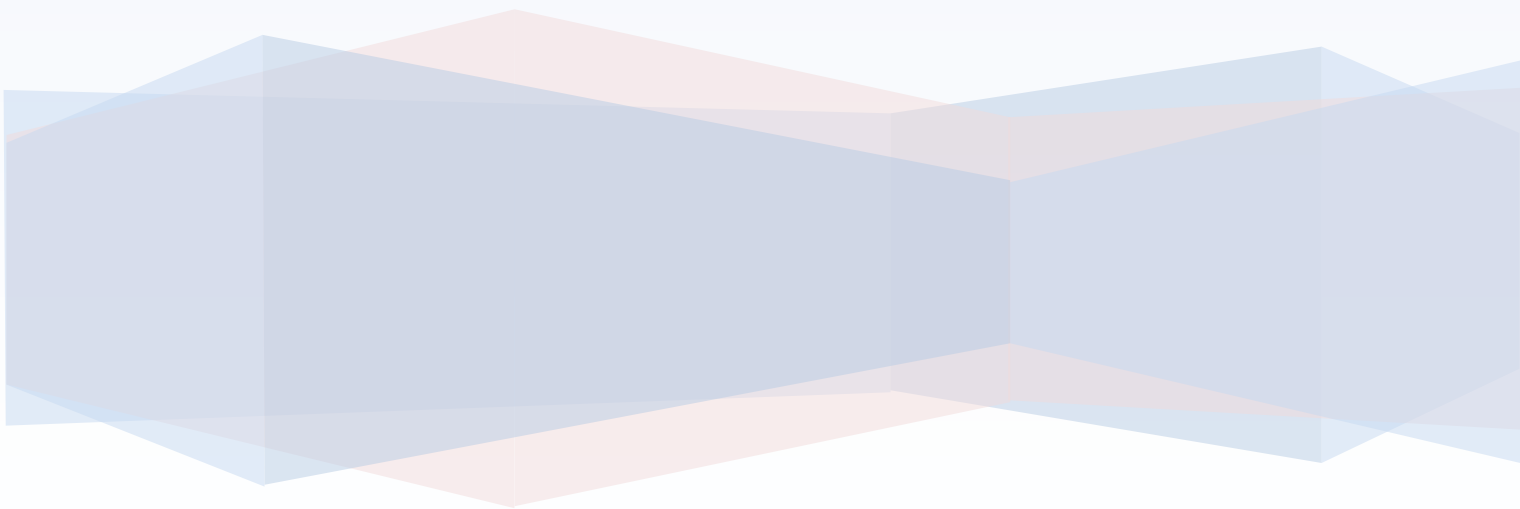


COS20015 – Fundamentals of Data Management

Distinction Report

Syed Muhammad Hassaan bin Ghayas 101231186



Introduction

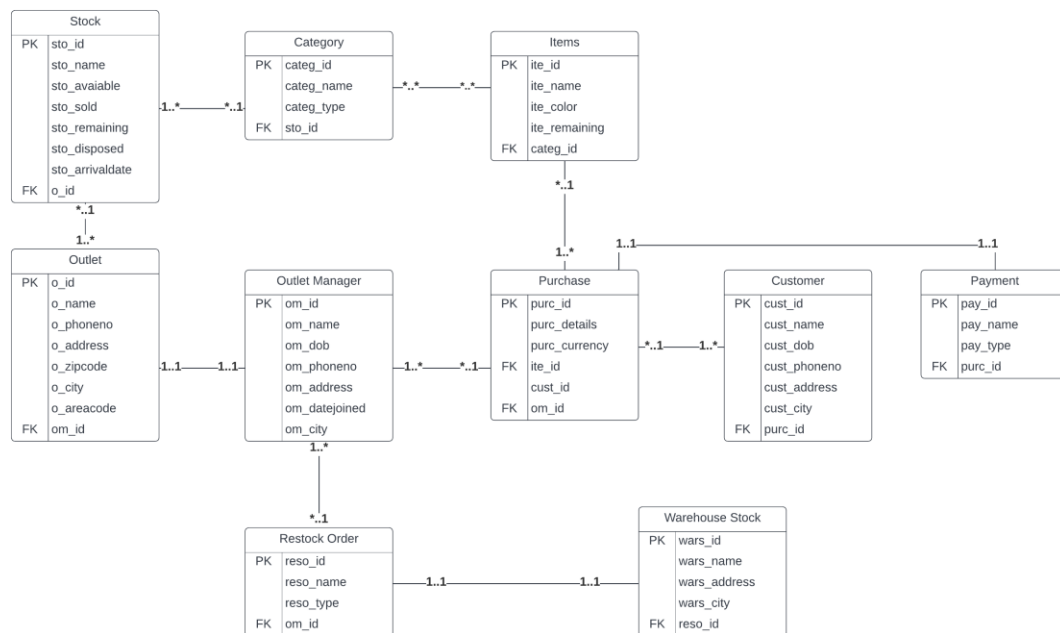
J. is a rich, elegant, and stylish fashion retail brand that deals exclusively in traditional ensembles. It regularly adds seasonal clothing, machine prints, intricate embroidery, and modern cuts to its already extensive collection. J. has quickly expanded its reach throughout Pakistan and internationally. Its 50+ outlets in Pakistan alone are proof of the company's efforts to become the country's largest fashion retail brand. Best of all, each location has a fresh and inspiring atmosphere in which customers can shop in comfort. J. also sells a wide variety of perfumes that can be worn by anyone. They range from light fresh flowery fragrances to heavy musk fragrances, all of which are extremely delectable and pleasant.

Overview of the System and its Business rules.

Without an effective IT infrastructure, no business can survive. One of the secrets to our success is our effective IT network. It is critical in providing us with information about what is going on both inside and outside the organisation. The Marketing and Sales Team is made up of experienced professionals who are well-versed in the market in which they operate. They are also well-informed about their surroundings and are constantly striving to keep the brand's market leadership position by collaborating with other departments. Finance The Finance Team makes all financial decisions, and their responsibility is to ensure that the company's financial statements are accurate and transparent.

J. brand has various outlets throughout Pakistan, with an Outlet Manager overseeing all purchases and stock records. When a customer makes a purchase, the database is divided into ten entities, each of which represents a purpose. The business model can be easily understood by looking at the ERD. When a purchase is made, the Outlet Manager and payment retrieve data and look for the item's category stock to see if it is available. If it is not available, the Outlet Manager sends a restock order to the Warehouse stocks so that all items can be delivered to the desired Outlet and business can run smoothly.

Entity Relationship diagram



Create Database:

SQL File 1*x

```

85  purc_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
86  purc_details VARCHAR(255) NOT NULL,
87  purc_currency VARCHAR(255) NOT NULL,
88  cust_id INT UNSIGNED NOT NULL,
89  ite_id INT UNSIGNED NOT NULL,
90  om_id INT UNSIGNED NOT NULL,
91  FOREIGN KEY(ite_id) REFERENCES Items(ite_id),
92  FOREIGN KEY(om_id) REFERENCES Outlet_Manager(om_id)
93  );
94
95
96
97  *
98  CREATE TABLE Customer(
99  cust_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
100  cust_name VARCHAR(255) NOT NULL,
101  cust_dob DATE NULL,
102  cust_phoneno VARCHAR(255) NOT NULL,
103  cust_address VARCHAR(255) NOT NULL,
104  cust_city VARCHAR(255) NOT NULL,
105  purc_id INT UNSIGNED NOT NULL,
106  FOREIGN KEY(purc_id) REFERENCES Purchase(purc_id)
107  );
108
109  *
110  CREATE TABLE Payment(
111  pay_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
112  pay_name VARCHAR(255) NOT NULL,
113  pay_type VARCHAR(255) NOT NULL,
114  purc_id INT UNSIGNED NOT NULL,
115  FOREIGN KEY(purc_id) REFERENCES Purchase(purc_id)
116  );
117
  
```

Action Output

	Time	Action	Message	Duration / Fetch
1	17:32:01	CREATE DATABASE DistinctionTask	1 row(s) affected	0.000 sec
2	17:32:01	USE DistinctionTask	0 row(s) affected	0.000 sec
3	17:32:01	CREATE TABLE Outlet_Manager(om_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, om_name VARCHAR(255) NOT NULL, om_dob DATE NULL, om_phoneno VARCHAR(255) NOT NULL, om_address VARCHAR(255) NOT NULL, om_datejoined DATE NULL, om_city VARCHAR(255) NOT NULL)	0 row(s) affected	0.020 sec
4	17:32:01	CREATE TABLE Restock_Order(reso_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, reso_name VARCHAR(255) NOT NULL, reso_type INT UNSIGNED NOT NULL)	0 row(s) affected	0.003 sec
5	17:32:01	CREATE TABLE Warehouse_Stock(wars_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, wars_name VARCHAR(255) NOT NULL, wars_address VARCHAR(255) NOT NULL, wars_city VARCHAR(255) NOT NULL)	0 row(s) affected	0.002 sec
6	17:32:01	CREATE TABLE Stock(sto_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, sto_name VARCHAR(255) NOT NULL, sto_available INT UNSIGNED NOT NULL, sto_sold INT UNSIGNED NOT NULL, sto_remaining INT UNSIGNED NOT NULL, sto_disposed INT UNSIGNED NOT NULL, sto_arrivaldate DATE NULL, o_id INT UNSIGNED NOT NULL)	0 row(s) affected	0.002 sec
7	17:32:01	CREATE TABLE Category(categ_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, categ_name VARCHAR(255) NOT NULL, categ_type INT UNSIGNED NOT NULL)	0 row(s) affected	0.002 sec
8	17:32:01	CREATE TABLE Items(ite_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, ite_name VARCHAR(255) NOT NULL, ite_color INT UNSIGNED NOT NULL, ite_remaining INT UNSIGNED NOT NULL, categ_id INT UNSIGNED NOT NULL)	0 row(s) affected	0.002 sec
9	17:32:01	CREATE TABLE Purchase(purc_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, purc_details VARCHAR(255) NOT NULL, purc_currency VARCHAR(255) NOT NULL, ite_id INT UNSIGNED NOT NULL, cust_id INT UNSIGNED NOT NULL, om_id INT UNSIGNED NOT NULL)	0 row(s) affected	0.002 sec
10	17:32:01	CREATE TABLE Customer(cust_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, cust_name VARCHAR(255) NOT NULL, cust_dob DATE NULL, cust_phoneno VARCHAR(255) NOT NULL, cust_address VARCHAR(255) NOT NULL, cust_city VARCHAR(255) NOT NULL)	0 row(s) affected	0.002 sec
11	17:32:01	CREATE TABLE Payment(pay_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, pay_name VARCHAR(255) NOT NULL, pay_type VARCHAR(255) NOT NULL, purc_id INT UNSIGNED NOT NULL)	0 row(s) affected	0.002 sec
12	17:32:01	CREATE TABLE Restock_Order(reso_id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY, reso_name VARCHAR(255) NOT NULL, reso_type INT UNSIGNED NOT NULL)	0 row(s) affected	0.002 sec

Insert Rows:

SQL File 1*x

```

121 (0, 'Outlet', 'PKR', 0,0,0),
122 (10, 'Outlet', 'PKR', 10,10,10);
123
124
125 • INSERT INTO Customer
126 (cust_id, cust_name, cust_dob, cust_phoneno, cust_address, cust_city, purc_id)
127 VALUES
128 (1, 'Azam', '1992-1-10', '01234567891', '01 North Town', 'Islamabad', 1),
129 (2, 'Zaid', '1993-2-9', '09876543212', '02 South Town', 'Karachi', 2),
130 (3, 'Sana', '1994-3-8', '09876543213', '03 East Town', 'Islamabad', 3),
131 (4, 'Ahmed', '1995-4-7', '09876543214', '04 West Town', 'Karachi', 4),
132 (5, 'Ibraheem', '1996-06-02', '09876543215', '05 West Town', 'Lahore', 5),
133 (6, 'Hassaan', '1997-6-5', '09876543216', '06 South Town', 'Islamabad', 6),
134 (7, 'Osama', '1998-7-4', '09876543217', '07 East Town', 'Karachi', 7),
135 (8, 'Abdullah', '1999-8-3', '09876543218', '08 North Town', 'Lahore', 8),
136 (9, 'Huzaifa', '2000-9-2', '09876543219', '09 West Town', 'Islamabad', 9),
137 (10, 'Khalid', '2001-06-02', '09876543100', '10 East View', 'Karachi', 10);
138
139
140 • INSERT INTO Payment
141 (pay_id, pay_name, pay_type, purc_id)
142 VALUES
143 (1, 'HBL', 'Card', 1),
144 (2, 'JazzCash', 'Mobile', 2),
145 (3, 'Haron', 'Cash', 3),

```

Action	Time	Action	Message	Duration / Fetch
1	17:32:56	USE DistinctionTask	0 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
2	17:32:56	INSERT INTO Outlet_Manager (om_id, om_name, om_dob, om_phono)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.002 sec
3	17:32:56	INSERT INTO Restock_Order (reso_id, reso_name, reso_type, reso_dob)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
4	17:32:56	INSERT INTO Warehouse_Stock (wars_id, wars_name, wars_dob)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
5	17:32:56	INSERT INTO Outlet (o_id, o_name, o_phoneno, o_address, o_city)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.001 sec
6	17:32:56	INSERT INTO Stock (sto_id, sto_name, sto_available, sto_sold, sto_dob)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.001 sec
7	17:32:56	INSERT INTO Category (categ_id, categ_name, categ_type, categ_dob)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.001 sec
8	17:32:56	INSERT INTO Items (ite_id, ite_name, ite_color, categ_id) VALUES	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.000 sec
9	17:32:56	INSERT INTO Purchase (purc_id, purc_details, purc_currency, purc_dob)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.001 sec
10	17:32:56	INSERT INTO Customer (cust_id, cust_name, cust_dob, cust_phono, cust_address, cust_city)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.001 sec
11	17:32:56	INSERT INTO Payment (pay_id, pay_name, pay_type, purc_id)	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.001 sec

Use Cases

1. It determines how many items are available in each category or how many items are remaining.

SCRIPT:

```
SELECT
c.categ_name AS CategoryName, c.categ_type AS CategoryType,
i.ite_remaining AS NumberOfItem
FROM
Category c
INNER JOIN Items i ON c.categ_id=i.categ_id
GROUP BY
c.categ_name, c.categ_type;
```

SCREENSHOT:

The screenshot displays a SQL query execution environment. The top pane shows the SQL script, and the bottom pane shows the results of the query.

SQL Script:

```
1 * SELECT
2 c.categ_name AS CategoryName, c.categ_type AS CategoryType, i.ite_remaining AS NumberOfItem
3 FROM
4 Category c
5 INNER JOIN Items i ON c.categ_id=i.categ_id
6 GROUP BY
7 c.categ_name, c.categ_type;
8
```

Result Set:

#	CategoryName	CategoryType	NumberOfItem
1	Cat-001-Core	Clothes	10
2	Cat-002-Musk	Perfume	20
3	Cat-003-Heaven	Clothes	30
4	Cat-004-khaddi	Perfume	40
5	Cat-005-Cotton	Clothes	50
6	Cat-006-Special	Perfume	40
7	Cat-007-Polo	Clothes	30
8	Cat-008-Empress	Clothes	20
9	Cat-009-Denim	Perfume	10
10	Cat-010-Medusa	Perfume	5

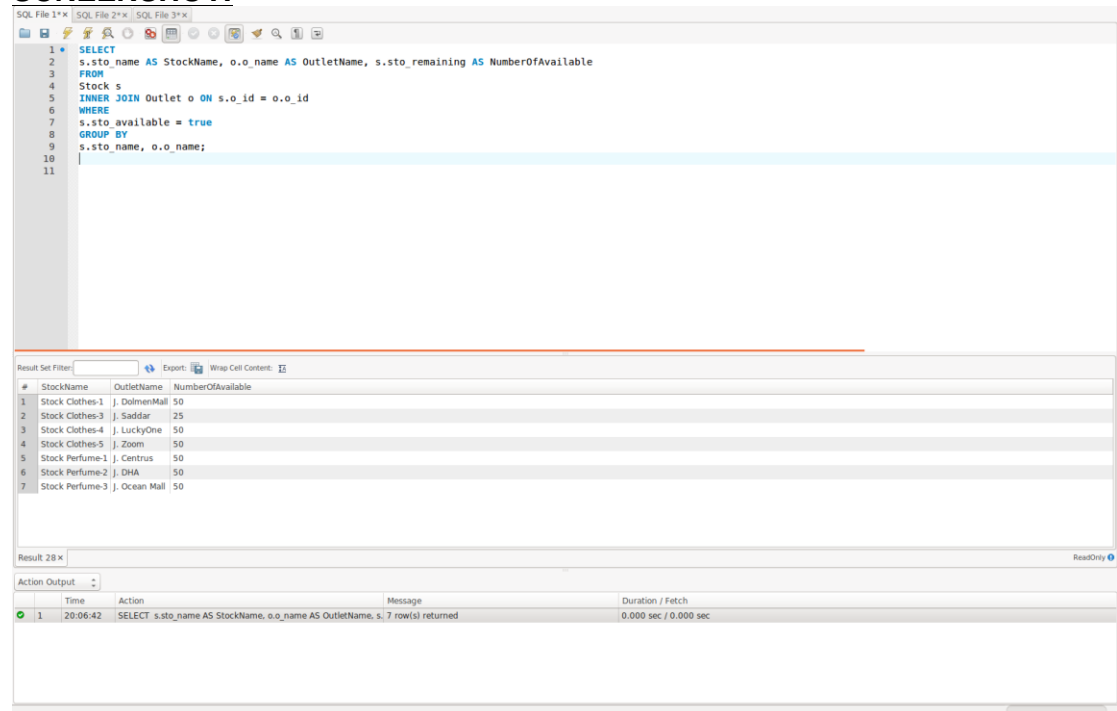
Action Output:

	Time	Action	Message	Duration / Fetch
1	19:37:24	SELECT c.categ_name AS CategoryName, c.categ_type AS CategoryType, i.ite_remaining AS NumberOfItem FROM Category c INNER JOIN Items i ON c.categ_id=i.categ_id GROUP BY c.categ_name, c.categ_type;	SELECT c.categ_name AS CategoryName, c.categ_type AS CategoryType, i.ite_remaining AS NumberOfItem FROM Category c INNER JOIN Items i ON c.categ_id=i.categ_id GROUP BY c.categ_name, c.categ_type; 10 row(s) returned	0.000 sec / 0.000 sec

2. Get the number of available stocks according to the Outlets, with StockName and OutletName.

SCRIPT:

```
SELECT
s.sto_name AS StockName, o.o_name AS OutletName, s.sto_remaining AS
NumberOfAvailable
FROM
Stock s
INNER JOIN Outlet o ON s.o_id = o.o_id
WHERE
s.sto_available = true
GROUP BY
s.sto_name, o.o_name;
```

SCREENSHOT:

The screenshot shows a SQL IDE interface with a script editor at the top containing the SQL query. Below the editor, the 'Result Set Filter' section displays the query results in a table format. The table has three columns: StockName, OutletName, and NumberOfAvailable. The results show 7 rows of data. At the bottom, the 'Action Output' section shows the execution details, including the time (20:06:42), the action (SELECT), the message (7 row(s) returned), and the duration (0.000 sec / 0.000 sec).

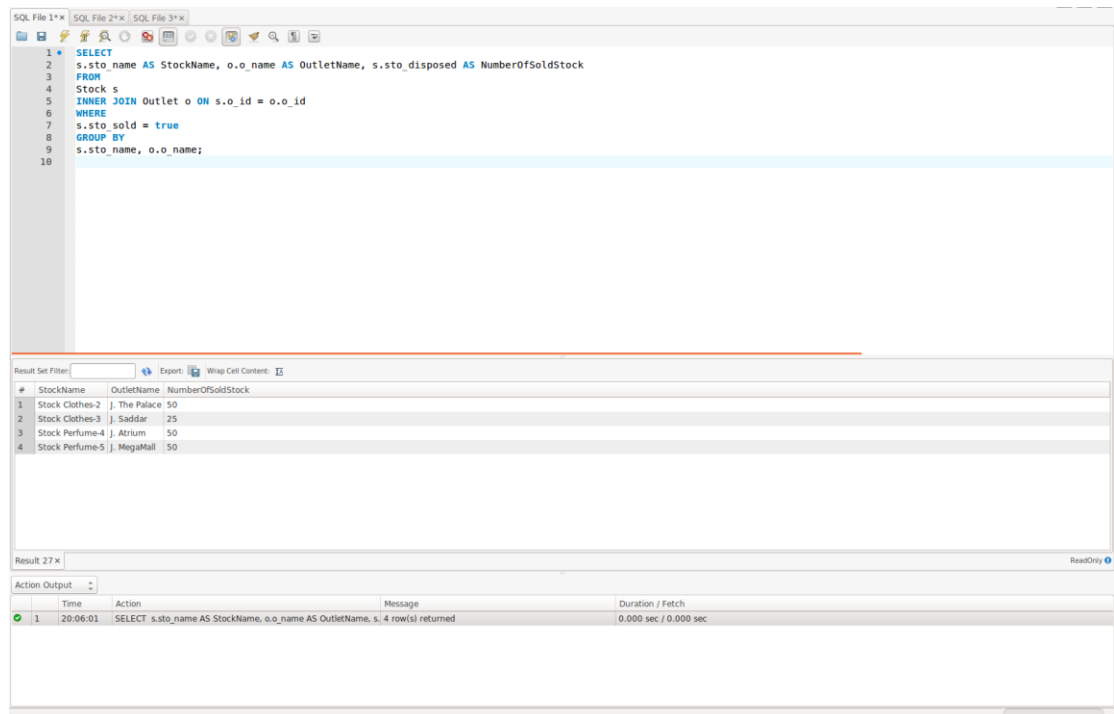
#	StockName	OutletName	NumberOfAvailable
1	Stock Clothes-1	J. DolmenMall	50
2	Stock Clothes-3	J. Saddar	25
3	Stock Clothes-4	J. LuckyOne	50
4	Stock Clothes-5	J. Zoom	50
5	Stock Perfume-1	J. Centrus	50
6	Stock Perfume-2	J. DHA	50
7	Stock Perfume-3	J. Ocean Mall	50

	Time	Action	Message	Duration / Fetch
1	20:06:42	SELECT	s.sto_name AS StockName, o.o_name AS OutletName, s. 7 row(s) returned	0.000 sec / 0.000 sec

3. Get the number of the sold stock from the outlet with Outlet Name and Stock Name

SCRIPT:

```
SELECT
s.sto_name AS StockName, o.o_name AS OutletName, s.sto_disposed AS
NumberOfSoldStock
FROM
Stock s
INNER JOIN Outlet o ON s.o_id = o.o_id
WHERE
s.sto_sold = true
GROUP BY
s.sto_name, o.o_name;
```

SCREENSHOT:

The screenshot shows a SQL IDE interface with a query editor at the top and a results pane at the bottom. The query editor contains the following SQL script:

```
1 SELECT
2 s.sto_name AS StockName, o.o_name AS OutletName, s.sto_disposed AS NumberOfSoldStock
3 FROM
4 Stock s
5 INNER JOIN Outlet o ON s.o_id = o.o_id
6 WHERE
7 s.sto_sold = true
8 GROUP BY
9 s.sto_name, o.o_name;
```

The results pane displays the output of the query, showing 4 rows of data. The columns are StockName, OutletName, and NumberOfSoldStock. The data is as follows:

#	StockName	OutletName	NumberOfSoldStock
1	Stock Clothes-2	J. The Palace	50
2	Stock Clothes-3	J. Saddam	25
3	Stock Perfume-4	J. Atrium	50
4	Stock Perfume-5	J. MegaMall	50

Below the results pane, there is an 'Action Output' section showing the execution details of the query:

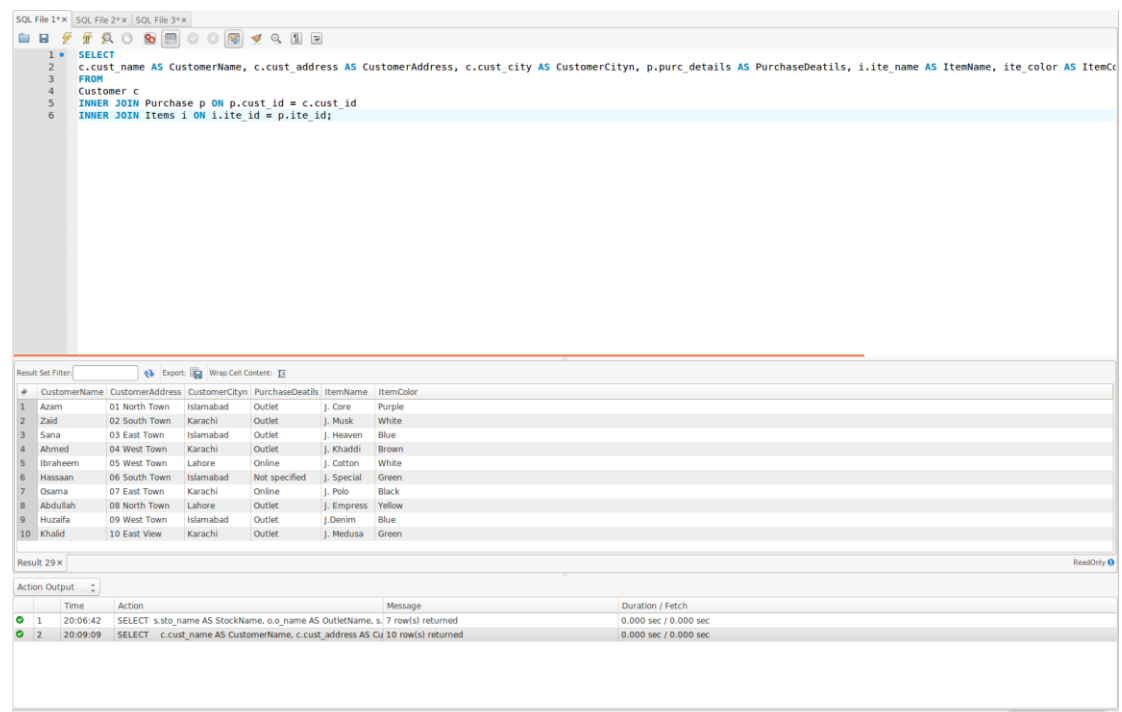
	Time	Action	Message	Duration / Fetch
1	20:06:01	SELECT s.sto_name AS StockName, o.o_name AS OutletName, s.sto_disposed AS NumberOfSoldStock	s: 4 row(s) returned	0.000 sec / 0.000 sec

4. List of customers who have bought items with item name, city, address and item colour. To deliver the product and rider can go and deliver the item with the respective colour and to accurate address.

SCRIPT:

```
SELECT
c.cust_name AS CustomerName, c.cust_address AS CustomerAddress,
c.cust_city AS CustomerCityn, p.purc_details AS PurchaseDeatils, i.ite_name
AS ItemName, ite_color AS ItemColor
FROM
Customer c
INNER JOIN Purchase p ON p.cust_id = c.cust_id
INNER JOIN Items i ON i.ite_id = p.ite_id;
```

SCREENSHOT:



The screenshot displays a SQL IDE interface with a query editor at the top and a results pane at the bottom. The query editor contains the following SQL script:

```
1 * SELECT
2 c.cust_name AS CustomerName, c.cust_address AS CustomerAddress, c.cust_city AS CustomerCityn, p.purc_details AS PurchaseDeatils, i.ite_name AS ItemName, ite_color AS ItemColor
3 FROM
4 Customer c
5 INNER JOIN Purchase p ON p.cust_id = c.cust_id
6 INNER JOIN Items i ON i.ite_id = p.ite_id;
```

The results pane shows a table with 10 rows of data. The columns are: #, CustomerName, CustomerAddress, CustomerCityn, PurchaseDeatils, ItemName, and ItemColor.

#	CustomerName	CustomerAddress	CustomerCityn	PurchaseDeatils	ItemName	ItemColor
1	Azam	01 North Town	Islamabad	Outlet	J. Core	Purple
2	Zaid	02 South Town	Karachi	Outlet	J. Musk	White
3	Sana	03 East Town	Islamabad	Outlet	J. Heaven	Blue
4	Ahmed	04 West Town	Karachi	Outlet	J. Khalidi	Brown
5	Ibraheem	05 West Town	Lahore	Online	J. Cotton	White
6	Hassaan	06 South Town	Islamabad	Not specified	J. Special	Green
7	Osama	07 East Town	Karachi	Online	J. Polo	Black
8	Abdullah	08 North Town	Lahore	Outlet	J. Empress	Yellow
9	Huzafa	09 West Town	Islamabad	Outlet	J. Denim	Blue
10	Khalid	10 East View	Karachi	Outlet	J. Medusa	Green

Below the results table, the 'Action Output' pane shows the execution details of the query:

	Time	Action	Message	Duration / Fetch
1	20:06:42	SELECT s.sto_name AS StockName, o.o_name AS OutletName, s. 7 row(s) returned		0.000 sec / 0.000 sec
2	20:09:09	SELECT c.cust_name AS CustomerName, c.cust_address AS Cu 10 row(s) returned		0.000 sec / 0.000 sec

5. This case is used to track client payment data with currency payment method and payment type.

SCRIPT:

```
SELECT
c.cust_name AS CustomerName, c.cust_address AS CustomerAddress,
c.cust_city AS CustomerCityn, p.purc_details AS PurchaseDeatils,
p.purc_currency AS Currency, py.pay_type AS PaymentType, py.pay_name
AS PaymentType
FROM
Customer c
INNER JOIN Purchase p ON p.cust_id = c.cust_id
INNER JOIN Payment py ON py.purc_id = p.purc_id;
```

SCREENSHOT:

SQL File 1* x SQL File 2* x SQL File 3* x

```
1 * SELECT
2 c.cust_name AS CustomerName, c.cust_address AS CustomerAddress, c.cust_city AS CustomerCityn, p.purc_details AS PurchaseDeatils, p.purc_currency AS Currency, py.pay_type AS
3 FROM
4 Customer c
5 INNER JOIN Purchase p ON p.cust_id = c.cust_id
6 INNER JOIN Payment py ON py.purc_id = p.purc_id;
```

Result Set Filter: Export Wrap Cell Content: 11

#	CustomerName	CustomerAddress	CustomerCityn	PurchaseDetails	Currency	PaymentType	PaymentType
1	Azam	01 North Town	Islamabad	Outlet	PKR	Card	HLB
2	Zaid	02 South Town	Karachi	Outlet	PKR	Mobile	jazzCash
3	Sana	03 East Town	Islamabad	Outlet	PKR	Cash	Haroon
4	Ahmed	04 West Town	Karachi	Outlet	PKR	Mobile	jazzCash
5	Ibraheem	05 West Town	Lahore	Online	USD	Card	Meezan Bank
6	Hassaan	06 South Town	Islamabad	Not specified	PKR	Mobile	Sadapay
7	Osama	07 East Town	Karachi	Online	AED	Card	Standard Charted
8	Abdullah	08 North Town	Lahore	Outlet	PKR	Cash	Nayapay
9	Huzaifa	09 West Town	Islamabad	Outlet	PKR	Mobile	jazzCash
10	Khalid	10 East View	Karachi	Outlet	PKR	Mobile	EasyPaiza

Result 31 x

Action Output

	Time	Action	Message	Duration / Fetch
1	20:11:59	SELECT c.cust_name AS CustomerName, c.cust_address AS CustomerAddress, c.cust_city AS CustomerCityn, p.purc_details AS PurchaseDeatils, p.purc_currency AS Currency, py.pay_type AS PaymentType, py.pay_name AS PaymentType	10 row(s) returned	0.000 sec / 0.000 sec

6. Get Outlet Manager name and His number who made sales to Customer with customer's PurchaseDetail and CustomerID.

SCRIPT:

```
SELECT
om.om_name AS OutletManager, om.om_phoneno AS PhoneNumber,
p.purc_details AS PurchaseDetail, p.cust_id AS CustomerID
FROM
Purchase p
INNER JOIN Outlet_Manager om ON om.om_id=p.om_id;
```

SCREENSHOT:

The screenshot shows a SQL query execution window. The query is as follows:

```
SELECT
om.om_name AS OutletManager, om.om_phoneno AS PhoneNumber, p.purc_details AS PurchaseDetail, p.cust_id AS CustomerID
FROM
Purchase p
INNER JOIN Outlet_Manager om ON om.om_id=p.om_id;
```

The results are displayed in a table with the following columns: #, OutletManager, PhoneNumber, PurchaseDetail, and CustomerID. The results are as follows:

#	OutletManager	PhoneNumber	PurchaseDetail	CustomerID
1	Umar	01111111111	Outlet	1
2	Wasim	02222222222	Outlet	2
3	Ahmed	03333333333	Outlet	3
4	Zaem	04444444444	Outlet	4
5	Sara	05555555555	Online	5
6	Sana	06666666666	Not specified	6
7	Zohaib	07777777777	Online	7
8	Zain	08888888888	Outlet	8
9	Haris	09999999999	Outlet	9
10	Farukh	01000000000	Outlet	10

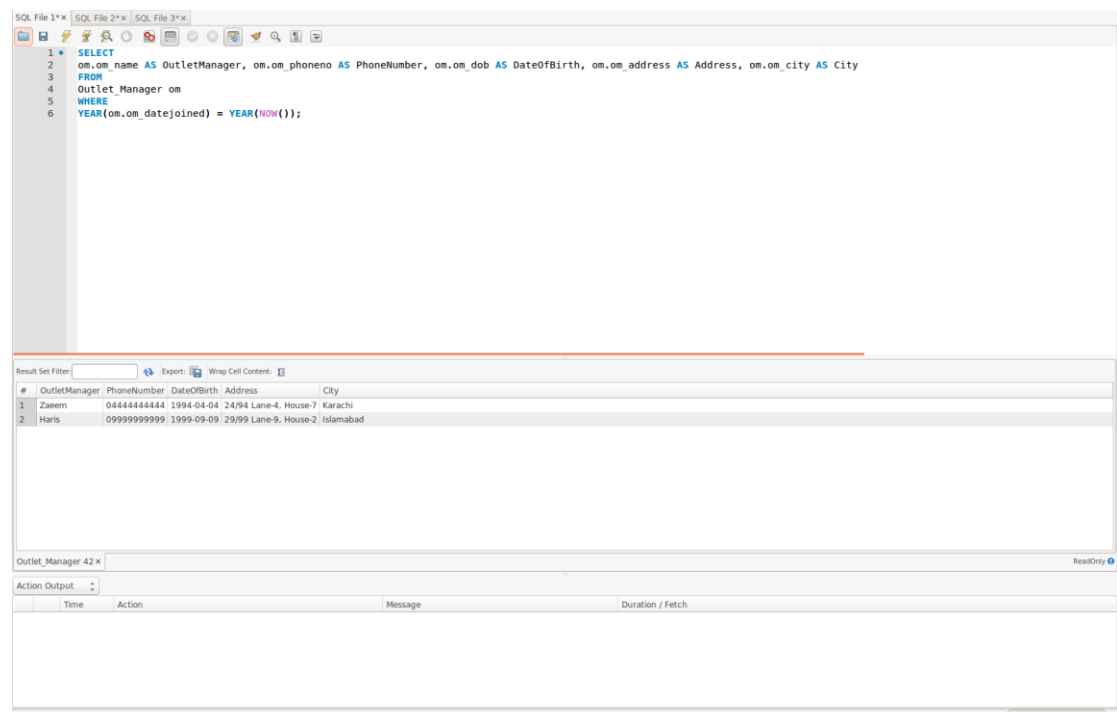
The Action Output section shows the following message:

Time	Action	Message	Duration / Fetch
20:23:07	SELECT	om.om_name AS OutletManager, om.om_phoneno AS (10 row(s) returned	0.000 sec / 0.000 sec

7. Get the list of Outlet Manager details who joined the Outlet that specific year with his Personal details such as PhoneNumber, DOB, Address and City.

SCRIPT:

```
SELECT
om.om_name AS OutletManager, om.om_phoneno AS PhoneNumber,
om.om_dob AS DateOfBirth, om.om_address AS Address, om.om_city AS
City
FROM
Outlet_Manager om
WHERE
YEAR(om.om_datejoined) = YEAR(NOW());
```

SCREENSHOT:

The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains the following SQL script:

```
1 SELECT
2 om.om_name AS OutletManager, om.om_phoneno AS PhoneNumber, om.om_dob AS DateOfBirth, om.om_address AS Address, om.om_city AS City
3 FROM
4 Outlet_Manager om
5 WHERE
6 YEAR(om.om_datejoined) = YEAR(NOW());
```

The results pane displays the output of the query, showing two rows of data:

#	OutletManager	PhoneNumber	DateOfBirth	Address	City
1	Zaem	04444444444	1994-04-04	24/94 Lane-4, House-7	Karachi
2	Haris	09999999999	1999-09-09	29/99 Lane-9, House-2	Islamabad

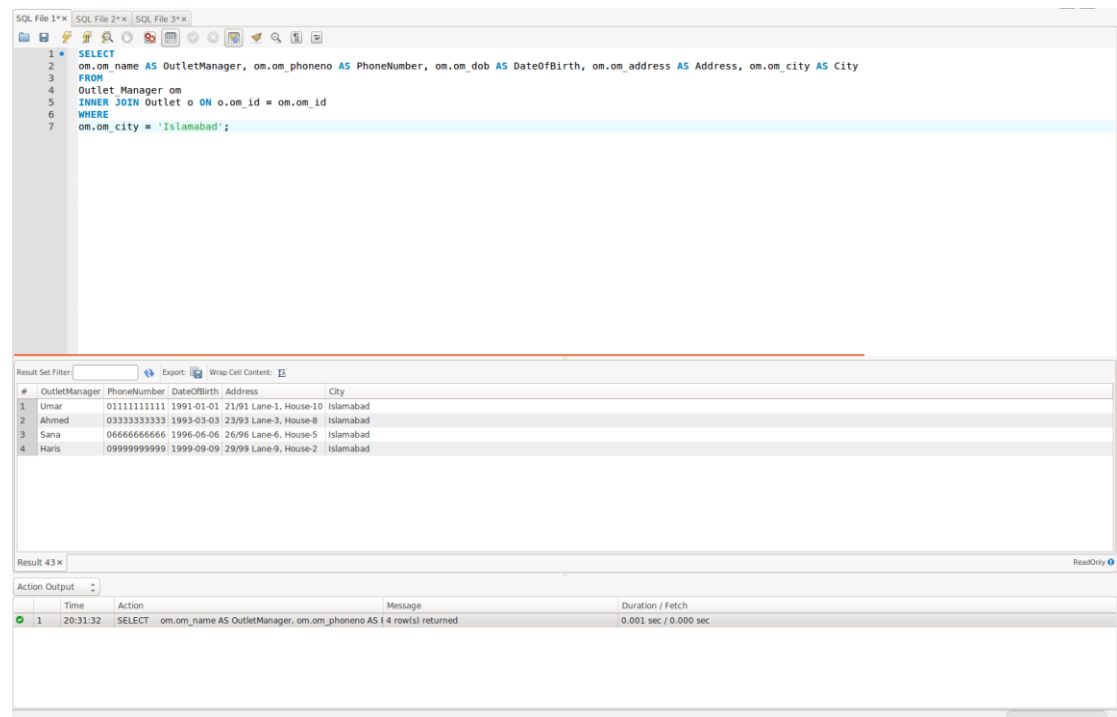
Below the results pane, there is a section for 'Action Output' with columns for Time, Action, Message, and Duration / Fetch.

8. This report shows Outlet Managers who live in Islamabad and who operate that outlet with his details like NAME, PhoneNumber, DOB, Address and City.

SCRIPT:

```
SELECT
om.om_name AS OutletManager, om.om_phoneno AS PhoneNumber,
om.om_dob AS DateOfBirth, om.om_address AS Address, om.om_city AS City
FROM
Outlet_Manager om
INNER JOIN Outlet o ON o.om_id = om.om_id
WHERE
om.om_city = 'Islamabad';
```

SCREENSHOT:



The screenshot shows a SQL IDE interface with a query editor at the top and a results pane at the bottom. The query editor contains the following SQL script:

```
1 * SELECT
2 om.om_name AS OutletManager, om.om_phoneno AS PhoneNumber, om.om_dob AS DateOfBirth, om.om_address AS Address, om.om_city AS City
3 FROM
4 Outlet_Manager om
5 INNER JOIN Outlet o ON o.om_id = om.om_id
6 WHERE
7 om.om_city = 'Islamabad';
```

The results pane displays the output of the query, showing 4 rows of data. The columns are: OutletManager, PhoneNumber, DateOfBirth, Address, and City. The data is as follows:

#	OutletManager	PhoneNumber	DateOfBirth	Address	City
1	Umar	01111111111	1991-01-01	21/91 Lane-1, House-10	Islamabad
2	Ahmed	03333333333	1993-03-03	23/93 Lane-3, House-8	Islamabad
3	Sana	06666666666	1996-06-06	26/96 Lane-6, House-5	Islamabad
4	Haris	09999999999	1999-09-09	29/99 Lane-9, House-2	Islamabad

Below the results pane, there is an 'Action Output' section showing the execution details of the query:

	Time	Action	Message	Duration / Fetch
1	20:31:32	SELECT	om.om_name AS OutletManager, om.om_phoneno AS f 4 row(s) returned	0.001 sec / 0.000 sec

9. Get the all-outlet manager who are related with respective ReStock data and Warehouse with all the essential details.

SCRIPT:

SELECT

om.om_name AS OutletManager, om.om_phoneno AS PhoneNumber,
om.om_dob AS DateOfBirth, om.om_address AS Address, om.om_city AS
City, ro.reso_name AS RestockName, ro.reso_type AS RestockType,
ws.wars_name AS WarehouseName, ws.wars_address AS
WarehouseAddress, ws.wars_city WarehouseCity

FROM

Outlet_Manager om

INNER JOIN Restock_Order ro ON ro.om_id = om.om_id

INNER JOIN Warehouse_Stock ws ON ws.reso_id = ro.reso_id;

SCREENSHOT:

SQL File 1* x SQL File 2* x SQL File 3* x

```

1 * SELECT
2 om.om_name AS OutletManager, om.om_phoneno AS PhoneNumber, om.om_dob AS DateOfBirth, om.om_address AS Address, om.om_city AS City, ro.reso_name AS RestockName, ro.reso_type
3 FROM
4 Outlet_Manager om
5 INNER JOIN Restock_Order ro ON ro.om_id = om.om_id
6 INNER JOIN Warehouse_Stock ws ON ws.reso_id = ro.reso_id;

```

Result Set Filter: Export Wrap Cell Content: 11

#	OutletManager	PhoneNumber	DateOfBirth	Address	City	RestockName	RestockType	WarehouseName	WarehouseAddress	WarehouseCity
1	Umar	01111111111	1991-01-01	21/91 Lane-1, House-10	Islamabad	StockNo-001	Perfume	South Point	Shop # 19, Gulshan-e-Ravi	Islamabad
2	Waqar	02222222222	1992-02-02	22/92 Lane-2, House-9	Karachi	StockNo-002	Perfume	East Point	6/7 House -53	Islamabad
3	Ahmed	03333333333	1993-03-03	23/93 Lane-3, House-8	Islamabad	StockNo-003	Perfume	Defense	Danapur Road, G.O.R.-1	Lahore
4	Zaem	04444444444	1994-04-04	24/94 Lane-4, House-7	Karachi	StockNo-004	Clothes	Bahria Town	4-Data Market	Lahore
5	Sara	05555555555	1995-05-05	25/95 Lane-5, House-6	Lahore	StockNo-005	Clothes	Bahria Town	C.P. & Barar Society Amir Khuro Rd, Dhoraji Colony	Karachi
6	Sara	06666666666	1996-06-06	26/96 Lane-6, House-5	Islamabad	StockNo-006	Ladies Clothes	DHA	Main Clifton Road	Karachi
7	Zohab	07777777777	1997-07-07	27/97 Lane-7, House-4	Karachi	StockNo-007	Shirt	Lakshmi Chowk	Shahrah-e-Fatima Jinnah	Lahore
8	Zain	08888888888	1998-08-08	28/98 Lane-8, House-3	Lahore	StockNo-008	Perfume	South Point	6/112 Rangpura Peer House	Islamabad
9	Haris	09999999999	1999-09-09	29/99 Lane-9, House-2	Islamabad	StockNo-009	Jeans	Saddar	Zainab Mkt., Nr. Khyber Pull	Karachi
10	Farukh	01000000000	2000-10-10	20/100 Lane-10, House-1	Lahore	StockNo-010	Perfume	East Point	Glamour One	Islamabad

Result 45 x

Action Output

Time	Action	Message	Duration / Fetch
20:35:55	SELECT om.om_name AS OutletManager, om.om_phoneno AS Ph	10 row(s) returned	0.000 sec / 0.000 sec

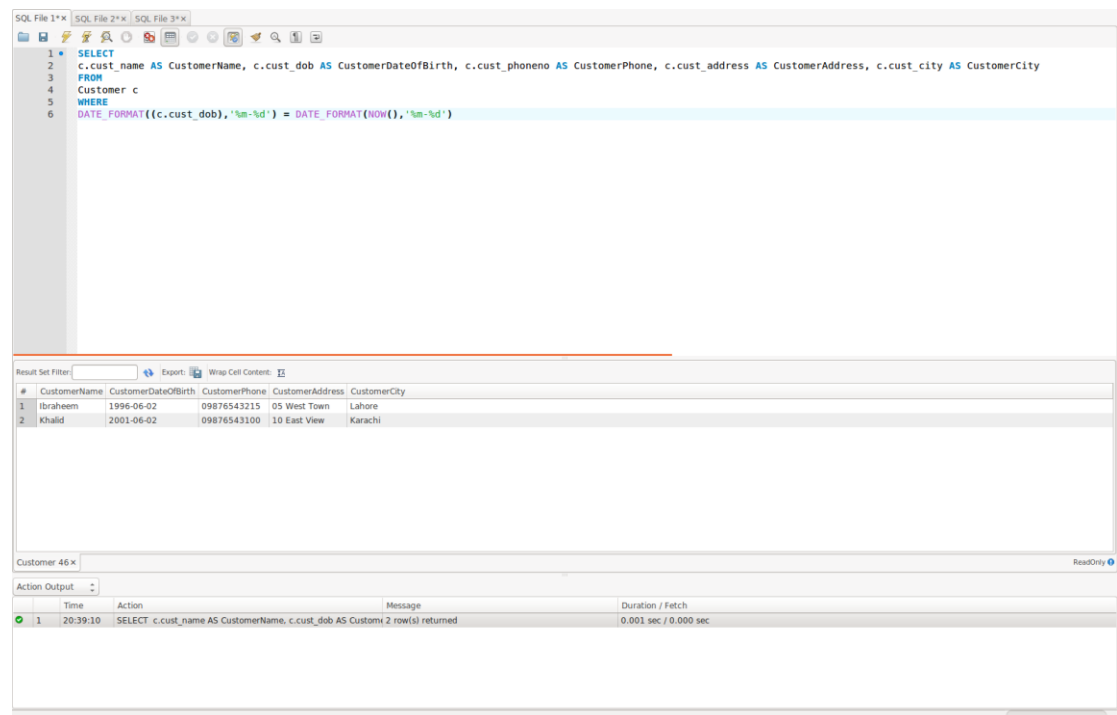
10. Get the list of customers who have birthday that particular date when data is retrieved (Today/Now).

This is to offer Customer discount on their happiest day to make it even more better.

SCRIPT:

```
SELECT
c.cust_name AS CustomerName, c.cust_dob AS CustomerDateOfBirth,
c.cust_phoneno AS CustomerPhone, c.cust_address AS CustomerAddress,
c.cust_city AS CustomerCity
FROM
Customer c
WHERE
DATE_FORMAT((c.cust_dob), '%m-%d') = DATE_FORMAT(NOW(), '%m-%d')
```

SCREENSHOT:



The screenshot shows a SQL IDE interface with a query editor at the top and a results pane at the bottom. The query editor contains the following SQL script:

```
1 SELECT
2 c.cust_name AS CustomerName, c.cust_dob AS CustomerDateOfBirth, c.cust_phoneno AS CustomerPhone, c.cust_address AS CustomerAddress, c.cust_city AS CustomerCity
3 FROM
4 Customer c
5 WHERE
6 DATE_FORMAT((c.cust_dob), '%m-%d') = DATE_FORMAT(NOW(), '%m-%d')
```

The results pane displays the output of the query, showing a table with 5 columns: CustomerName, CustomerDateOfBirth, CustomerPhone, CustomerAddress, and CustomerCity. The table contains 2 rows of data:

#	CustomerName	CustomerDateOfBirth	CustomerPhone	CustomerAddress	CustomerCity
1	Ibraheem	1996-06-02	09876543215	05 West Town	Lahore
2	Khalid	2001-06-02	09876543100	10 East View	Karachi

Below the results pane, there is an 'Action Output' section showing the execution details:

	Time	Action	Message	Duration / Fetch
1	20:39:10	SELECT c.cust_name AS CustomerName, c.cust_dob AS CustomerDateOfBirth, c.cust_phoneno AS CustomerPhone, c.cust_address AS CustomerAddress, c.cust_city AS CustomerCity FROM Customer c WHERE DATE_FORMAT((c.cust_dob), '%m-%d') = DATE_FORMAT(NOW(), '%m-%d')	2 row(s) returned	0.001 sec / 0.000 sec

Conclusion

This database is a prototype that the company may use. More items can be added to make it more detailed and interesting. This database best reflects the J. brand that I highlighted in my proposal. This database will assist the Company in extracting information more effectively by utilising the aforementioned scenarios.

This work gave me with all of the necessary information, and I am now able to enter and use cases to get data. This report could be improved by adding more entries to it; I met all of the requirements despite having an issue with my electricity; I could have made this report even better, but it is still meeting all of the standards for a Distinction. I wish I had more time to attempt the Higher Distinction task, But I did the great Job looking at the database and cases looking at the time period I invested in this Distinction task.