



ISM 6217 – Database Management Systems

Project – Part 3 5 points

JASHWANTH DASARI (Z23761602)

Sree Teja Reddy Chappidi (Z23754920)

Task: Data manipulation on the tables created in Project – Part 2.

Note: Present this on a separate Word document named Project – Part 3.

Perform the following functions using the following commands **on any of the tables created** in Project Part 2. Paste SQL and results for each. For the UPDATE command perform Select * and provide a screenshot of the results.

Explain in at least ‘one’ sentence the business problem statement for ‘each’ of these queries.

Please do not use all these commands on similar tables. You need to diversify so that all your tables get involved in this assignment.

1. UPDATE using WHERE (ampersand (&) in the clause) - *0.5 points*

Oracle SQL Developer: ISM 6217-DASARI

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

ISM 6217-DASARI

Tables (Filtered)

Views

Indexes

Packages

Procedures

Functions

Operators

Queues

Queues Tables

Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Editions

Java

XML Schemas

XML DB Repository

OLAP Option

Analytic Views

Scheduler

Property Graph

Reports

All Reports

Analytic View Reports

Data Dictionary Reports

Data Modeler Reports

OLAP Reports

Worksheet

Query Builder

```
-- Select all data from invoice_info
SELECT *
FROM invoice_info;

-- Update the due date for a specific invoice
UPDATE invoice_info
SET due_date = '2023-07-30'
WHERE invoice_number = '&INV-2023-001';

-- Select all data from invoice_info
SELECT *
FROM invoice_info;
```

Script Output

ScriptRunner Task

Error starting at line : 6 in command

```
UPDATE invoice_info
SET due_date = '2023-07-30'
WHERE invoice_number = 'INV-2023-001'
```

Error report -

ORA-01861: literal does not match format string

INVOICE_NUMBER	INVOICE_D	DUE_DATE	PATIENT_NUMBER	DOCTOR_ID
INV-2023-001	15-JUN-23	15-JUL-23	5618	1467

Enter Substitution Variable

Enter value for 2023:

OK Cancel

Line 9 Column 1 | Insert | Modified | Windows: C

Oracle SQL Developer: ISM 6217-DASARI

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

ISM 6217-DASARI

Tables (Filtered)

Views

Indexes

Packages

Procedures

Functions

Operators

Queues

Queues Tables

Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Editions

Java

XML Schemas

XML DB Repository

OLAP Option

Analytic Views

Scheduler

Property Graph

Reports

All Reports

Analytic View Reports

Data Dictionary Reports

Data Modeler Reports

Worksheet

Query Builder

```
-- Select all data from invoice_info
SELECT *
FROM invoice_info;

-- Update the due date for a specific invoice
UPDATE invoice_info
SET due_date = '2023-07-30'
WHERE invoice_number = '&INV-2023-001';

-- Select all data from invoice_info
SELECT *
FROM invoice_info;
```

Script Output

ScriptRunner Task

Enter Substitution Variable

Enter value for INV:

INV-2023-001

OK Cancel

INVOICE_NUMBER	INVOICE_D	DUE_DATE	PATIENT_NUMBER	DOCTOR_ID
INV-2023-001	15-JUN-23	15-JUL-23	5618	1467

Type here to search

11:54 PM 23-Oct-24

Oracle SQL Developer: ISM 6217-DASARI

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

ISM 6217-DASARI

Tables (Filtered)

Views

Indexes

Packages

Procedures

Functions

Operators

Queues

Queues Tables

Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Editions

Java

XML Schemas

XML DB Repository

OLAP Option

Analytic Views

Scheduler

Property Graph

Reports

All Reports

Analytic View Reports

Data Dictionary Reports

Data Modeler Reports

OLAP Reports

Worksheet

Query Builder

```
-- Select all data from invoice_info
SELECT *
FROM invoice_info;

-- Update the due date for a specific invoice
UPDATE invoice_info
SET due_date = '&due_date'
WHERE invoice_number = '&INV-2023-001';

-- Select all data from invoice_info
SELECT *
FROM invoice_info;
```

Script Output

ScriptRunner Task

Enter Substitution Variable

Enter value for Due_date:

2023-07-30

OK Cancel

INVOICE_NUMBER	INVOICE_D	DUE_DATE	PATIENT_NUMBER	DOCTOR_ID
INV-2023-001	15-JUN-23	15-JUL-23	5618	1467

Line 9 Column 1 | Insert | Modified | Windows: C

Oracle SQL Developer: ISM 6217-DASARI

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

ISM 6217-DASARI

Tables (Filtered)

Views

Indexes

Packages

Procedures

Functions

Operators

Queues

Queues Tables

Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

Directories

Editions

Java

XML Schemas

XML DB Repository

OLAP Option

Analytic Views

Scheduler

Property Graph

Reports

All Reports

Analytic View Reports

Data Dictionary Reports

Data Modeler Reports

OLAP Reports

Worksheet

Query Builder

```
-- Select all data from invoice_info
SELECT *
FROM invoice_info;

-- Update the due date for a specific invoice
UPDATE invoice_info
SET due_date = '&due_date'
WHERE invoice_number = '&INV-2023-001';

-- Select all data from invoice_info
SELECT *
FROM invoice_info;
```

Script Output

ScriptRunner Task

Enter Substitution Variable

Enter value for Due_date:

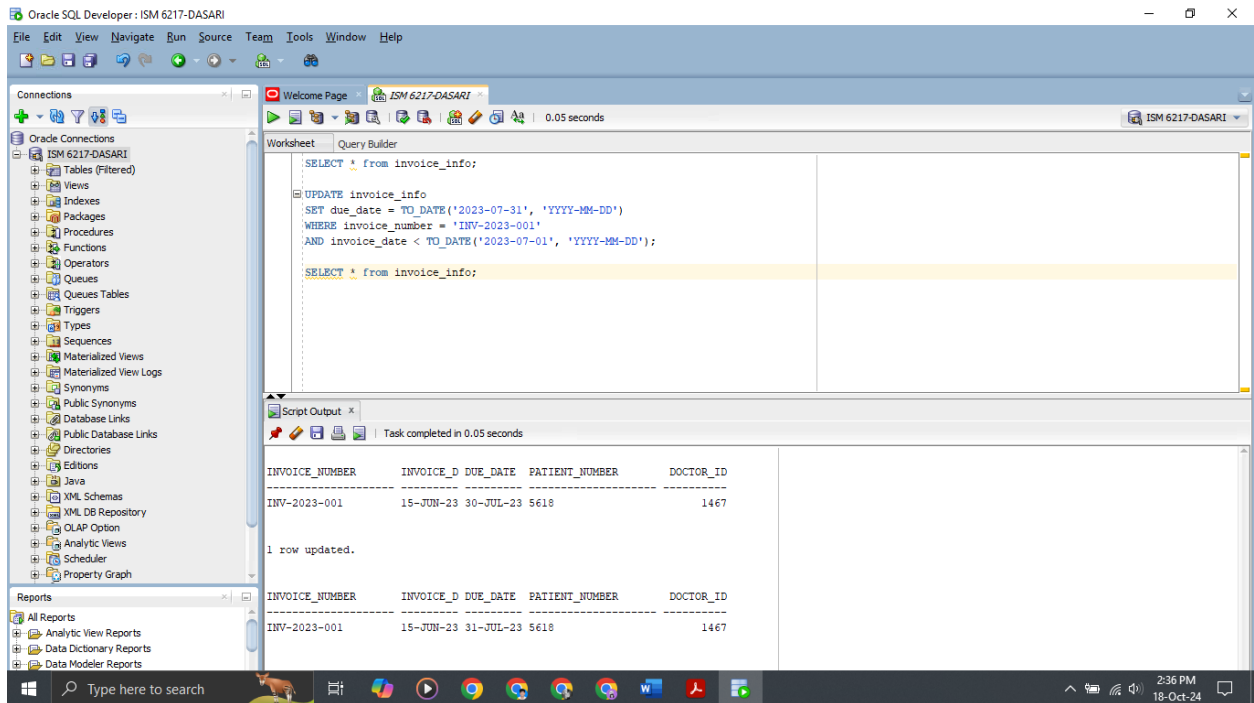
2023-07-31

OK Cancel

INVOICE_NUMBER	INVOICE_D	DUE_DATE	PATIENT_NUMBER	DOCTOR_ID
INV-2023-001	15-JUN-23	15-JUL-23	5618	1467

Type here to search

11:51 PM 23-Oct-24



I got for 1st time and again I was getting error for that

Business Problem Explanation:

This query updates the due date of invoices generated before a certain date, which could be useful for managing delayed payments or customer requests to extend the deadline.

2. SELECT using ROUND function – *0.5 points*

The screenshot shows the Oracle SQL Developer interface with a query in the Worksheet pane. The query is:

```
SELECT * FROM price_info;

SELECT invoice_number, code, qty, ROUND(total, 0) AS rounded_total
FROM price_info;
```

The Script Output pane shows the results of the query, which is completed in 0.042 seconds. The results are displayed in two tables:

INVOICE_NUMBER	CODE	QTY	UNIT_PRICE	TOTAL
INV-2023-001	CN001	1	10000	10000
INV-2023-001	SRV002	1	15000	15000
INV-2023-001	RX003	3	2000	6000

INVOICE_NUMBER	CODE	QTY	ROUNDED_TOTAL
INV-2023-001	CN001	1	10000
INV-2023-001	SRV002	1	15000
INV-2023-001	RX003	3	6000

Business Problem Explanation:

Rounding the total price to the nearest whole number can be useful when preparing customer invoices or reports that do not need to include fractional amounts, improving readability.

3. NON-EQUALITY join of two tables using the JOIN...ON – *0.5 points*

The screenshot shows the Oracle SQL Developer interface with a query in the Worksheet pane. The query is:

```
SELECT * FROM price_info;
SELECT * FROM service_info;

SELECT p.invoice_number, p.code, p.total, s.unit_price, s.description
FROM price_info p
JOIN service_info s
ON p.total > s.unit_price;
```

The Script Output pane shows the results of the query, which is completed in 0.058 seconds. The results are displayed in three tables:

INVOICE_NUMBER	CODE	QTY	UNIT_PRICE	TOTAL
INV-2023-001	CN001	1	10000	10000
INV-2023-001	SRV002	1	15000	15000
INV-2023-001	RX003	3	2000	6000

CODE	UNIT	UNIT_PRICE	DESCRIPTION
SRV002	Service	15000	Laboratory Tests
CN001	Consultation	10000	General check-up
RX003	Prescription	2000	Medication

INVOICE_NUMBER	CODE	TOTAL	UNIT_PRICE	DESCRIPTION
INV-2023-001	SRV002	15000	10000	General check-up
INV-2023-001	SRV002	15000	2000	Medication
INV-2023-001	CN001	10000	2000	Medication
INV-2023-001	RX003	6000	2000	Medication

Business Problem Explanation:

This query identifies services where the total price charged on an invoice is greater than the base unit price. This can help spot cases where additional fees, taxes, or services were added to the base service cost.

4. OUTER JOIN of two tables (LEFT or RIGHT join) – 1 point

The screenshot shows the Oracle SQL Developer interface with a query window. The query is a LEFT JOIN between invoice_info and patient_info. The script output shows the results of the query.

```
SELECT * FROM invoice_info;
SELECT * FROM patient_info;

SELECT i.invoice_number, i.invoice_date, p.patient_name, p.patient_address
FROM invoice_info i
LEFT JOIN patient_info p
ON i.patient_number = p.patient_number;
```

INVOICE_NUMBER	INVOICE_D	DUE_DATE	PATIENT_NUMBER	DOCTOR_ID
INV-2023-001	15-JUN-23	31-JUL-23	5618	1467

PATIENT_NUMBER	PATIENT_NAME
5618	Sarah Johnson

INVOICE_NUMBER	INVOICE_D	PATIENT_NAME
INV-2023-001	15-JUN-23	Sarah Johnson

The screenshot shows the Oracle SQL Developer interface with a query window. The query is a RIGHT JOIN between invoice_info and patient_info. The query result shows the results of the query.

```
SELECT * FROM invoice_info;
SELECT * FROM patient_info;

SELECT i.invoice_number, i.invoice_date, p.patient_name, p.patient_address
FROM invoice_info i
RIGHT JOIN patient_info p
ON i.patient_number = p.patient_number;
```

INVOICE_NUMBER	INVOICE_D	DUE_DATE	PATIENT_NUMBER	DOCTOR_ID
INV-2023-001	15-JUN-23	15-JUL-23	5618	1467

PATIENT_NUMBER	PATIENT_NAME
5618	Sarah Johnson

INVOICE_NUMBER	INVOICE_D	PATIENT_NAME
INV-2023-001	15-JUN-23	Sarah Johnson

Business Problem Explanation:

This query helps identify invoices that may be missing associated patient details. This could be useful for data validation, ensuring that all invoices are properly attributed to patients.

5. Create a new table (copy of the existing table from your project) – then UPDATE one row of the table and perform MERGE to keep them in sync – *2.5 points*. Show me the SELECT * of both tables after copying. Then another SELECT * after UPDATE. Then another SELECT * on both the tables after MERGE.

- Create a copy of the doctor_info table:

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left lists the database 'ISM 6217-DASARI' and its tables. The 'Worksheet' pane in the center contains the following SQL script:

```
SELECT * FROM doctor_info;

CREATE TABLE doctor_info_copy AS
SELECT * FROM doctor_info;

SELECT * FROM doctor_info;
```

The 'Query Result' pane at the bottom displays the output of the queries. It shows the initial state of the 'doctor_info' table with one record, followed by a message indicating the creation of the 'doctor_info_copy' table, and then the same record in the new table.

DOCTOR_ID	DOCTOR_NAME	CLINIC_NAME
1467	Joe Black	Healthy Clinic

Table DOCTOR_INFO_COPY created.

DOCTOR_ID	DOCTOR_NAME	CLINIC_NAME
1467	Joe Black	Healthy Clinic

- SELECT all records from both tables to show the initial state:

Oracle SQL Developer: ISM 6217-DASARI

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

ISM 6217-DASARI

Tables (Filtered)

- ALL_SALES
- ALL_SALES2
- ALL_SALES3
- AUTHOR
- BOOKAUTHOR
- BOOKS
- COUPONS
- CUSTOMERS
- CUSTOMERS_1LDB
- DIVISIONS
- DOCTOR_INFO
- EMPLOYEES
- EMPLOYEES2
- EMPLOYEES3
- INVOICE_INFO
- JOBS
- JY_EMPLOYEES
- MORE_EMPLOYEES
- MORE_PRODUCTS
- NEW_CATEGORY
- NEW3
- ORDER_STATUS
- ORDERITEMS_1LDB
- ORDERS_1LDB
- PATIENT_INFO

Reports

All Reports

Analytic View Reports

Data Dictionary Reports

Data Modeler Reports

Worksheet

Query Builder

```
SELECT * FROM doctor_infor;
SELECT * FROM doctor_info_copy;
```

Script Output

Task completed in 0.099 seconds

DOCTOR_ID	DOCTOR_NAME	CLINIC_NAME
1467	Joe Black	Healthy Clinic
1467	Joe Black	Healthy Clinic

Type here to search

3:01 AM 19-Oct-24

- UPDATE a row in the copy:

Oracle SQL Developer: ISM 6217-DASARI

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

ISM 6217-DASARI

Tables (Filtered)

- ALL_SALES
- ALL_SALES2
- ALL_SALES3
- AUTHOR
- BOOKAUTHOR
- BOOKS
- COUPONS
- CUSTOMERS
- CUSTOMERS_1LDB
- DIVISIONS
- DOCTOR_INFO
- EMPLOYEES
- EMPLOYEES2
- EMPLOYEES3
- INVOICE_INFO
- JOBS
- JY_EMPLOYEES
- MORE_EMPLOYEES
- MORE_PRODUCTS
- NEW_CATEGORY
- NEW3
- ORDER_STATUS
- ORDERITEMS_1LDB
- ORDERS_1LDB
- PATIENT_INFO

Reports

All Reports

Analytic View Reports

Data Dictionary Reports

Data Modeler Reports

Worksheet

Query Builder

```
UPDATE doctor_info_copy
SET clinic_name = 'Dasari Jashwanth Clinic'
WHERE doctor_id = 1467;
```

Script Output

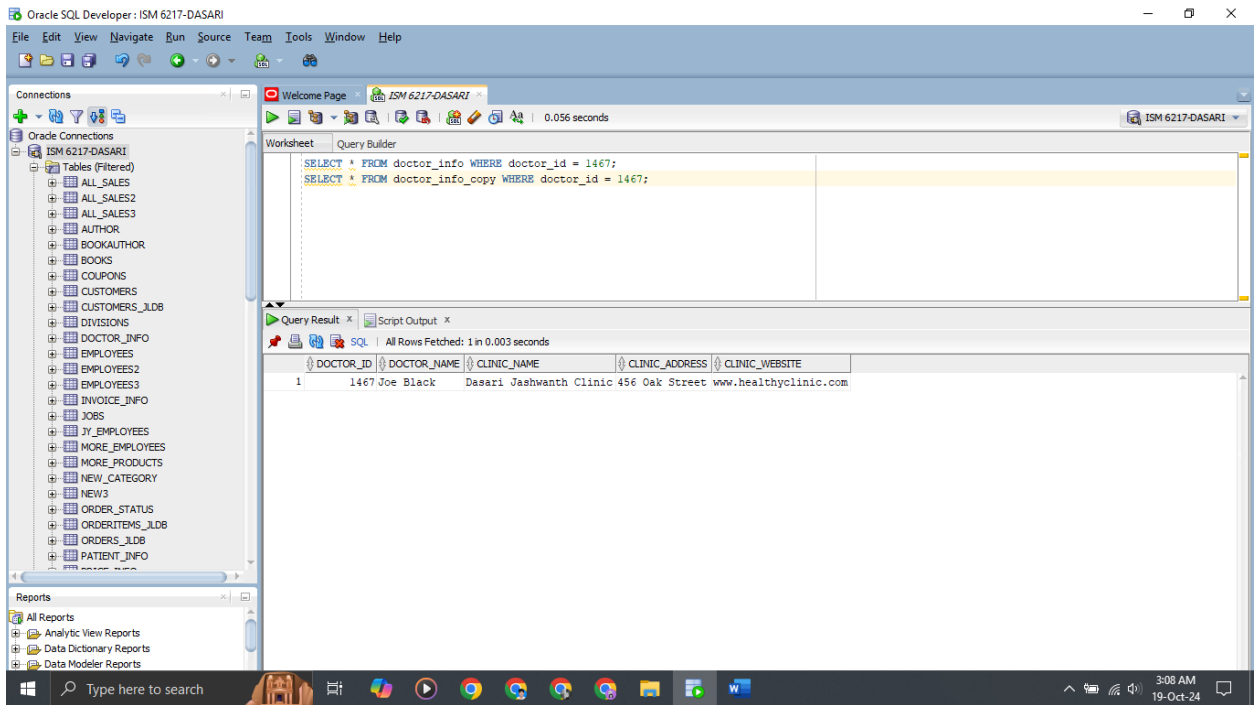
Task completed in 0.023 seconds

1 row updated.

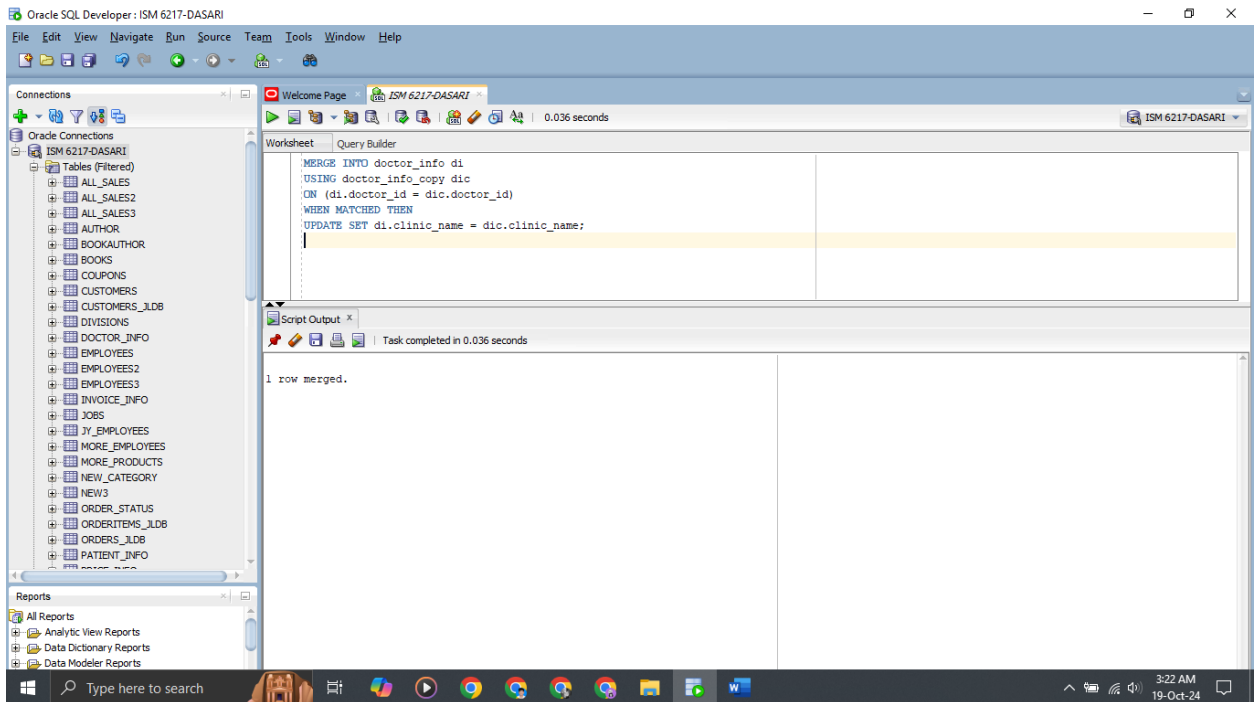
Type here to search

3:06 AM 19-Oct-24

- SELECT all records from both tables after the update:



- MERGE to sync the doctor_info table with the updated copy:



- SELECT all records from both tables after the MERGE:

The first screenshot shows the Oracle SQL Developer interface with a query in the Worksheet:

```
SELECT * FROM doctor_info;
SELECT * FROM doctor_info_copy;
```

The Script Output window shows the results of the query, displaying two rows of data:

DOCTOR_ID	DOCTOR_NAME	CLINIC_NAME
1467	Joe Black	Dasari Jashwanth Clinic
1467	Joe Black	Dasari Jashwanth Clinic

The second screenshot shows the same interface with a different query in the Worksheet:

```
SELECT * FROM doctor_info_copy;
```

The Script Output window shows the results of the query, displaying one row of data:

DOCTOR_ID	DOCTOR_NAME	CLINIC_NAME	CLINIC_ADDRESS	CLINIC_WEBSITE
1	1467 Joe Black	Dasari Jashwanth Clinic	456 Oak Street	www.healthyclinic.com

Business Problem Explanation:

This query ensures that a doctor's updated clinic information is reflected in the main table, maintaining consistency between the live data and a temporary copy. This approach can be used to stage updates or changes before pushing them to production.

