المملكة العربية السعودية
وزارة التعليم
جامعة جدة
كلية علوم و هندسة الحاسب

The *Hotel* Management System

Data Base

2410710	لارا تركي عسيري
2410570	نور كامل موريا
2410037	حنين عبيد القريقري
2410447	ديالا أحمد باسلامه

Table Of Contents:

Description:

The Hotel Management System is a complete solution designed to handle different parts of hotel operations smoothly.

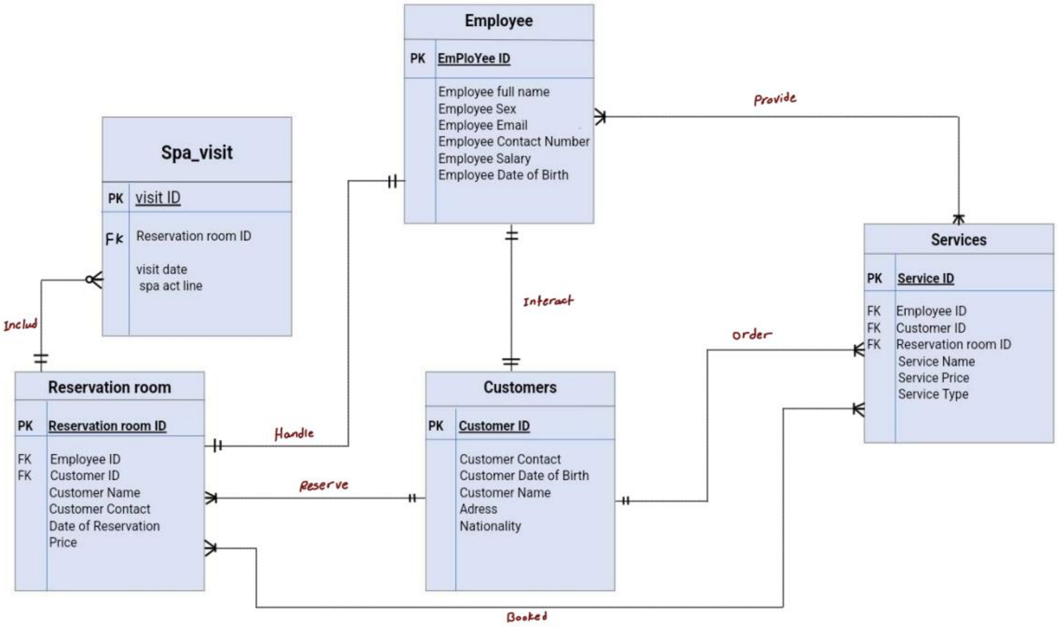
It covers service offerings, employee management, customer interactions, and reservation handling.

The Entities that will be used are:

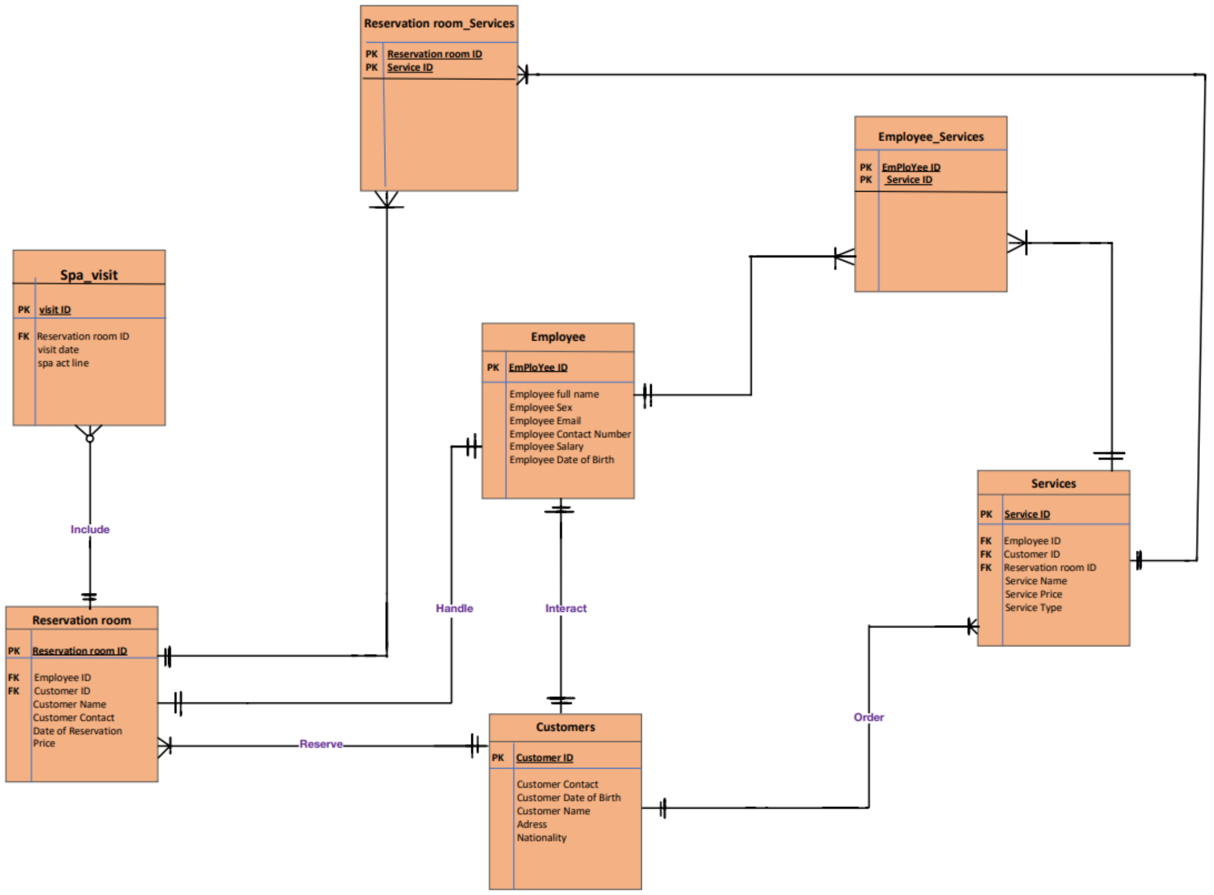
- EMPLOEE
- SERVICES
- COUSTUMERS
- RESERVATION_ROOM
- SPA_VISIT

Task	Team Members
➢ Design the database following an ER model and ERD .	لارا تركي عسيري
➢ Convert your ER diagram to a relational schema. ➢ Normalize the tables (each table should be in 3NF at least).	نور كامل موريا
➢ Use Oracle to create the normalized tables ➢ Populate your tables with 5 rows at least.	حنين عبيد القريقري
➢ Design and implement at least 4 queries. ➢ Design two stored procedures as the following ○ Create a PARAMETER based SELECT QUERY stored procedure which return recordsbased on parameters**?** ○ Create an UPDATE query based stored procedure.	ديالا باسلامه

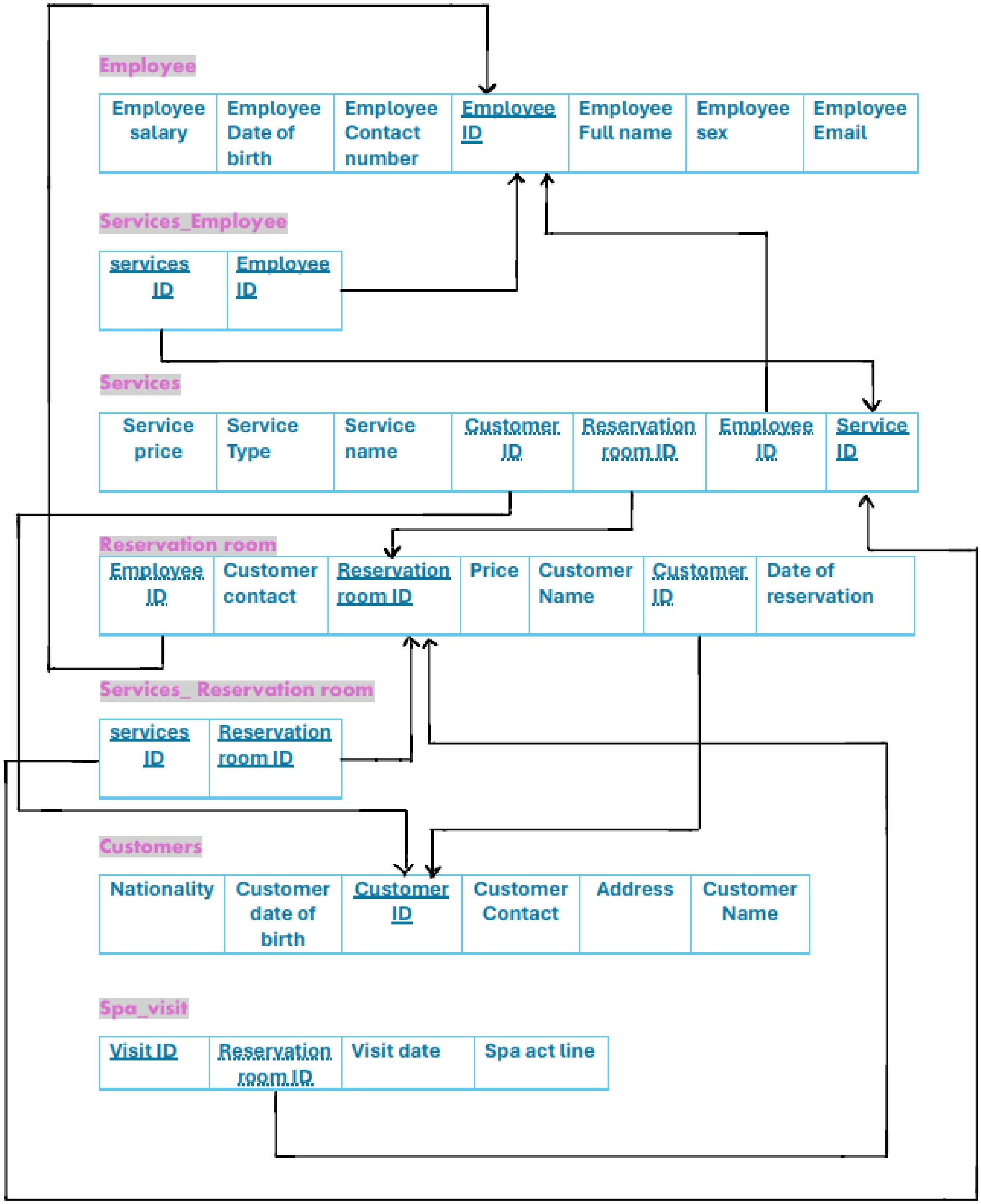
ER model:


Employee
PK
EmPloYee ID
Employee full name
Employee Sex
Employee Email
Employee Contact Number
Employee Salary
Employee Date of Birth
Spa_visit
PK
visit ID
Fk
Reservation room ID
visit date
spa act line
Services
PK
Service ID
FK
FK
FK
Employee ID
Customer ID
Reservation room ID
Service Name
Service Price
Service Type
Reservation room
PK
Reservation room ID
FK
FK
Employee ID
Customer ID
Customer Name
Customer Contact
Date of Reservation
Price
Customers
PK
Customer ID
Customer Contact
Customer Date of Birth
Customer Name
Adress
Nationality
Provide
Includ
Interact
order
Handle
Reserve
Booked

ERD model:


Reservation room_Services
PK Reservation room ID
PK Service ID
Employee_Services
PK EmPloYee ID
PK Service ID
Spa_visit
PK visit ID
FK Reservation room ID
visit date
spa act line
Employee
PK EmPloYee ID
Employee full name
Employee Sex
Employee Email
Employee Contact Number
Employee Salary
Employee Date of Birth
Services
PK Service ID
FK Employee ID
FK Customer ID
FK Reservation room ID
Service Name
Service Price
Service Type
Include
Handle
Interact
Order
Reservation room
PK Reservation room ID
FK Employee ID
FK Customer ID
Customer Name
Customer Contact
Date of Reservation
Price
Reserve
Customers
PK Customer ID
Customer Contact
Customer Date of Birth
Customer Name
Adress
Nationality

Relational Schema:



Employee

Employee salary	Employee Date of birth	Employee Contact number	Employee ID	Employee Full name	Employee sex	Employee Email

Services_Employee

services ID	Employee ID

Services

Service price	Service Type	Service name	Customer ID	Reservation room ID	Employee ID	Service ID

Reservation room

Employee ID	Customer contact	Reservation room ID	Price	Customer Name	Customer ID	Date of reservation

Services_ Reservation room

services ID	Reservation room ID

Customers

Nationality	Customer date of birth	Customer ID	Customer Contact	Address	Customer Name

Spa_visit

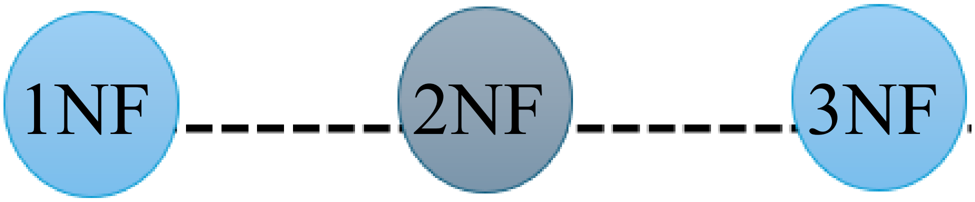
Visit ID	Reservation room ID	Visit date	Spa act line

Normalization:

(First Normal Form)

➢ No multivalued attributes.

➢ Every attribute value is atomic .



(Second Normal Form)

➢ Every non-key attribute is fully functionally dependent on the ENTIRE primary key .

➢ No partial functional dependencies .

Normalization:

(Third Normal Form)

➢ no transitive dependencies

(functional dependencies on non-primary-key attributes).

EMPLOYEE(Employee ID ,Employee salary, Employee Date of birth , Employee, Contact number, Employee, Full name , Employee sex ,Employee Email).

In third form because no transitive dependencies (functional dependencies on non-primary-key attributes) and No partial functional dependencies .

every non-key attribute is fully functionally dependent on the primary key(**Employee ID**) .No multivalued attributes and Every attribute value is atomic.

SERVICES_EMPLOYEE(services ID, Employe ID)

In third form because no transitive dependencies (functional dependencies on non-primary-key attributes) . No partial functional dependencies and no multivalued attributes and Every attribute value is atomic.

SERVICES(Service ID , Employee ID, Reservation room ID , Customer ID,

Service name, Service Type , Service price)

In third form because no transitive dependencies (functional dependencies on non-primary-key attributes) .Also, No partial functional dependencies

.every non-key attribute is fully functionally dependent on the primary key(Service ID) .

No multivalued attributes and Every attribute value is atomic.

RESERVATION_ROOM(Reservation room ID, Customer ID,

Employee ID,Customer contact, Price, CustomerName, Date of reservation)

In third form because no transitive dependencies (functional dependencies on non-primary-key attributes) . No partial functional dependencies . every non-key attribute is fully functionally dependent on the primary key(Reservation room ID) .

No multivalued attributes and Every attribute value is atomic.

SERVICES_ RESERVATION_ROOM (services ID, Reservation room ID)

In third form because no transitive dependencies (functional dependencies on non-primary-key attributes) . No partial functional dependencies .

No multivalued attributes and Every attribute value is atomic.

CUSTOMERS (Customer ID , Customer date of birth, Customer Contact, Nationality, Address, Customer Name)

In third form because no transitive dependencies (functional dependencies on non-primary-key attributes) . No partial functional dependencies .

every non-key attribute is fully functionally dependent on the primary key(Customer ID) .

No multivalued attributes and Every attribute value is atomic.

SPA_VISIT (Visit ID , Reservation room ID , Visit date , Spa act line)

In third form because no transitive dependencies (functional dependencies on non-primary-key attributes) .No partial functional dependencies .every non-key attribute is fully functionally dependent on the primary key(Visit ID) .

No multivalued attributes and Every attribute value is atomic.

Functional Dependencies (FD):

Employee ID → (Employee salary, Employee Date of birth , Employee, Contact number, Employee, Full name , Employee sex ,Employee Email).

Service ID → (Service name, Service Type , Service price).

Reservation room ID → (Customer contact, Price, CustomerName, Date of reservation).

Customer ID →(Customer date of birth, Customer Contact, Nationality, Address, Customer Name).

Visit ID → (Visit date , Spa act line).

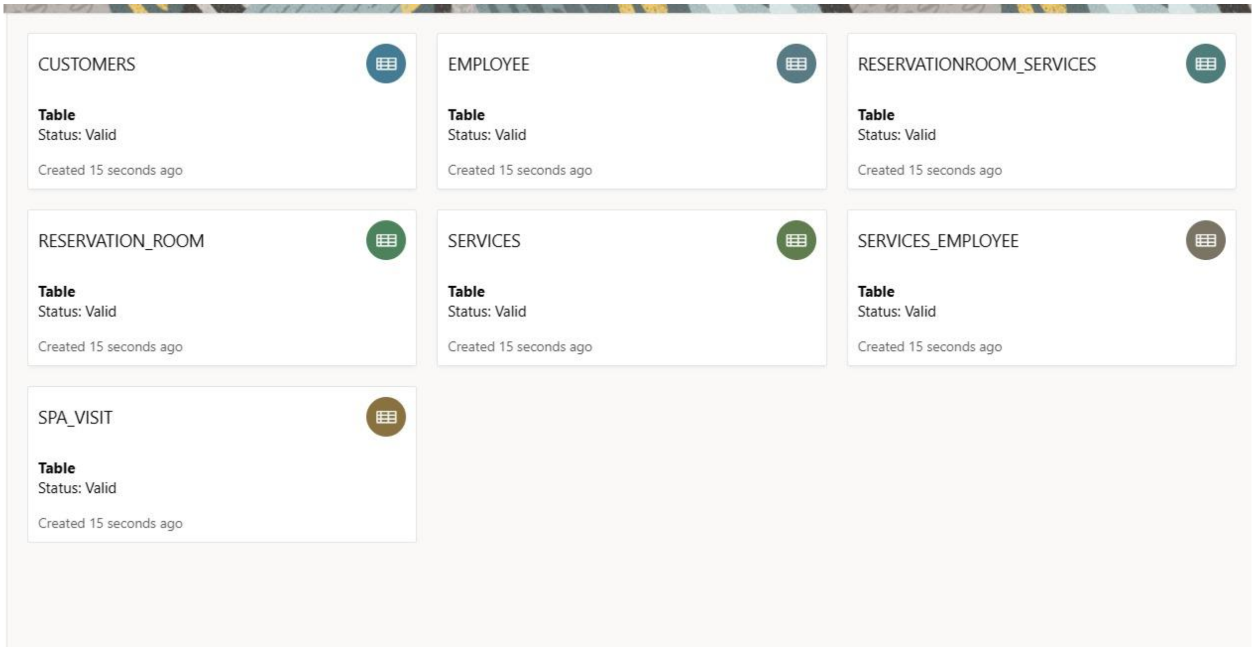
SQL :

- CREATE

```
-- Create CUSTOMERS table
CREATE TABLE CUSTOMERS (
Customer_ID INT PRIMARY KEY,
Customer_Date_of_Birth DATE,
Customer_Contact VARCHAR2(20),
Nationality VARCHAR2(50),
Address VARCHAR2(200),
Customer_Name VARCHAR2(100)
);
------------------------------------------------------------------------------------------------------------------------
-- Create EMPLOYEE table
CREATE TABLE EMPLOYEE (
Employee_ID INT PRIMARY KEY,
Employee_Salary DECIMAL(10, 2),
Employee_Date_of_Birth DATE,
Employee_Contact_Number VARCHAR2(20),
Employee_Full_Name VARCHAR2(100),
Employee_Sex CHAR(1),
Employee_Email VARCHAR2(100)
);

------------------------------------------------------------------------------------------------------------------------

-- Create RESERVATION_ROOM table
CREATE TABLE RESERVATION_ROOM (
Reservation_Room_ID INT PRIMARY KEY,
Customer_ID INT,
Employee_ID INT,
Customer_Contact VARCHAR2(20),
Price DECIMAL(10, 2),
Customer_Name VARCHAR2(100),
Date_of_Reservation DATE,
FOREIGN KEY (Customer_ID) REFERENCES CUSTOMERS(Customer_ID),
FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEE(Employee_ID)
);
```

```
-- Create SERVICES table
CREATE TABLE SERVICES (
Service_ID INT PRIMARY KEY,
Employee_ID INT,
Reservation_Room_ID INT,
Customer_ID INT,
Service_Name VARCHAR2(100),
Service_Type VARCHAR2(50),
Service_Price DECIMAL(10, 2),
FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEE(Employee_ID),
FOREIGN KEY (Reservation_Room_ID) REFERENCES
RESERVATION_ROOM(Reservation_Room_ID),
FOREIGN KEY (Customer_ID) REFERENCES CUSTOMERS(Customer_ID)
);

------------------------------------------------------------------------------------------------------------------------

-- Create RESERVATIONROOM_SERVICES table
CREATE TABLE RESERVATIONROOM_SERVICES (
Services_ID INT,
Reservation_Room_ID INT,
PRIMARY KEY (Services_ID, Reservation_Room_ID),
FOREIGN KEY (Services_ID) REFERENCES SERVICES(Service_ID),
FOREIGN KEY (Reservation_Room_ID) REFERENCES
RESERVATION_ROOM(Reservation_Room_ID)
);

------------------------------------------------------------------------------------------------------------------------

-- Create SERVICES_EMPLOYEE table
CREATE TABLE SERVICES_EMPLOYEE (
Services_ID INT PRIMARY KEY,
Employee_ID INT,
FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEE(Employee_ID)
);

------------------------------------------------------------------------------------------------------------------------
```

```
-- Create SPA_VISIT table
CREATE TABLE SPA_VISIT (
Visit_ID INT PRIMARY KEY,
Reservation_Room_ID INT,
Visit_Date DATE,
Spa_Act_Line VARCHAR2(200),
FOREIGN KEY (Reservation_Room_ID) REFERENCES
RESERVATION_ROOM(Reservation_Room_ID)
);
```



CUSTOMERS
Table
Status: Valid
Created 15 seconds ago

EMPLOYEE
Table
Status: Valid
Created 15 seconds ago

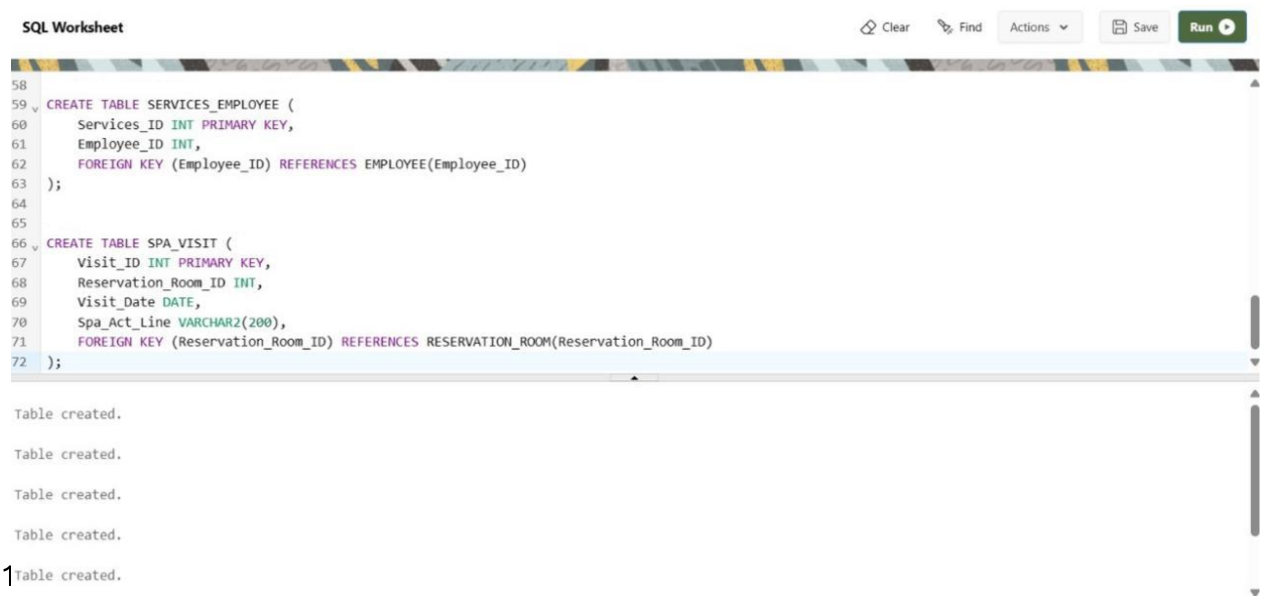
RESERVATIONROOM_SERVICES
Table
Status: Valid
Created 15 seconds ago

RESERVATION_ROOM
Table
Status: Valid
Created 15 seconds ago

SERVICES
Table
Status: Valid
Created 15 seconds ago

SERVICES_EMPLOYEE
Table
Status: Valid
Created 15 seconds ago

SPA_VISIT
Table
Status: Valid
Created 15 seconds ago

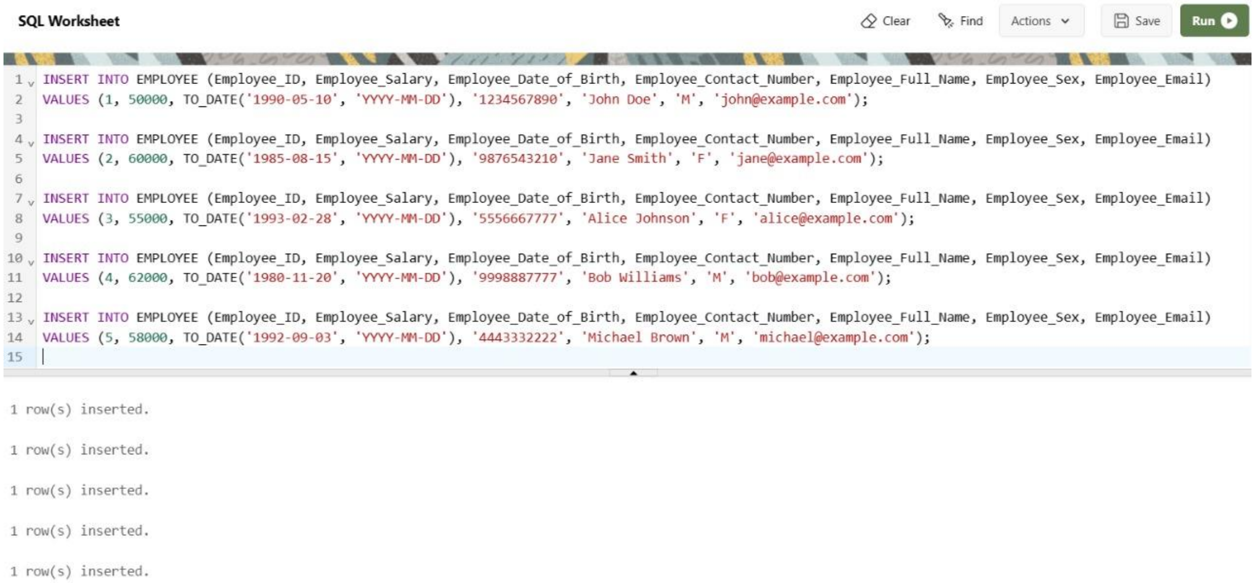


```
CREATE TABLE SERVICES_EMPLOYEE (
    Services_ID INT PRIMARY KEY,
    Employee_ID INT,
    FOREIGN KEY (Employee_ID) REFERENCES EMPLOYEE(Employee_ID)
);


CREATE TABLE SPA_VISIT (
    Visit_ID INT PRIMARY KEY,
    Reservation_Room_ID INT,
    Visit_Date DATE,
    Spa_Act_Line VARCHAR2(200),
    FOREIGN KEY (Reservation_Room_ID) REFERENCES RESERVATION_ROOM(Reservation_Room_ID)
);
```

```
Table created.

Table created.

Table created.

Table created.

Table created.
```

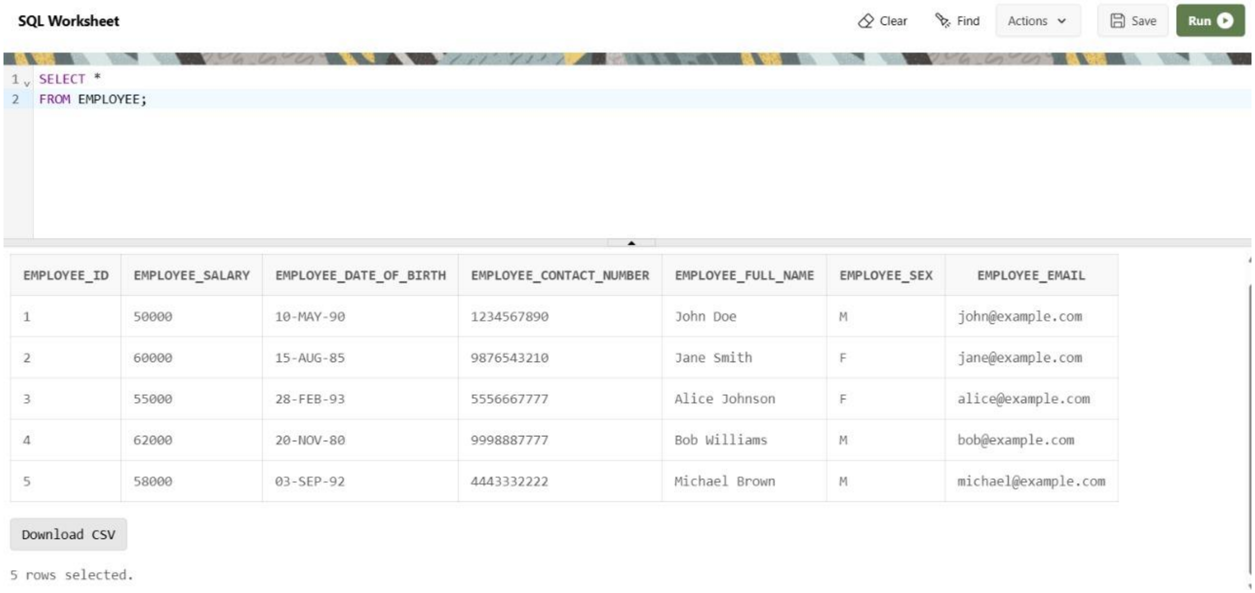
- INSERT

```
INSERT INTO EMPLOYEE (Employee_ID, Employee_Salary,
Employee_Date_of_Birth, Employee_Contact_Number, Employee_Full_Name,
Employee_Sex, Employee_Email)
VALUES (1, 50000, TO_DATE('1990-05-10', 'YYYY-MM-DD'), '1234567890', 'John
Doe', 'M', 'john@example.com');

INSERT INTO EMPLOYEE (Employee_ID, Employee_Salary,
Employee_Date_of_Birth, Employee_Contact_Number, Employee_Full_Name,
Employee_Sex, Employee_Email)
VALUES (2, 60000, TO_DATE('1985-08-15', 'YYYY-MM-DD'), '9876543210', 'Jane
Smith', 'F', 'jane@example.com');

INSERT INTO EMPLOYEE (Employee_ID, Employee_Salary,
Employee_Date_of_Birth, Employee_Contact_Number, Employee_Full_Name,
Employee_Sex, Employee_Email)
VALUES (3, 55000, TO_DATE('1993-02-28', 'YYYY-MM-DD'), '5556667777', 'Alice
Johnson', 'F', 'alice@example.com');

INSERT INTO EMPLOYEE (Employee_ID, Employee_Salary,
Employee_Date_of_Birth, Employee_Contact_Number, Employee_Full_Name,
Employee_Sex, Employee_Email)
VALUES (4, 62000, TO_DATE('1980-11-20', 'YYYY-MM-DD'), '9998887777', 'Bob
Williams', 'M', 'bob@example.com');

INSERT INTO EMPLOYEE (Employee_ID, Employee_Salary,
Employee_Date_of_Birth, Employee_Contact_Number, Employee_Full_Name,
Employee_Sex, Employee_Email)
VALUES (5, 58000, TO_DATE('1992-09-03', 'YYYY-MM-DD'), '4443332222', 'Michael
Brown', 'M', 'michael@example.com');
```



```sql
INSERT INTO EMPLOYEE (Employee_ID, Employee_Salary, Employee_Date_of_Birth, Employee_Contact_Number, Employee_Full_Name, Employee_Sex, Employee_Email)
VALUES (1, 50000, TO_DATE('1990-05-10', 'YYYY-MM-DD'), '1234567890', 'John Doe', 'M', 'john@example.com');

INSERT INTO EMPLOYEE (Employee_ID, Employee_Salary, Employee_Date_of_Birth, Employee_Contact_Number, Employee_Full_Name, Employee_Sex, Employee_Email)
VALUES (2, 60000, TO_DATE('1985-08-15', 'YYYY-MM-DD'), '9876543210', 'Jane Smith', 'F', 'jane@example.com');

INSERT INTO EMPLOYEE (Employee_ID, Employee_Salary, Employee_Date_of_Birth, Employee_Contact_Number, Employee_Full_Name, Employee_Sex, Employee_Email)
VALUES (3, 55000, TO_DATE('1993-02-28', 'YYYY-MM-DD'), '5556667777', 'Alice Johnson', 'F', 'alice@example.com');

INSERT INTO EMPLOYEE (Employee_ID, Employee_Salary, Employee_Date_of_Birth, Employee_Contact_Number, Employee_Full_Name, Employee_Sex, Employee_Email)
VALUES (4, 62000, TO_DATE('1980-11-20', 'YYYY-MM-DD'), '9998887777', 'Bob Williams', 'M', 'bob@example.com');

INSERT INTO EMPLOYEE (Employee_ID, Employee_Salary, Employee_Date_of_Birth, Employee_Contact_Number, Employee_Full_Name, Employee_Sex, Employee_Email)
VALUES (5, 58000, TO_DATE('1992-09-03', 'YYYY-MM-DD'), '4443332222', 'Michael Brown', 'M', 'michael@example.com');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

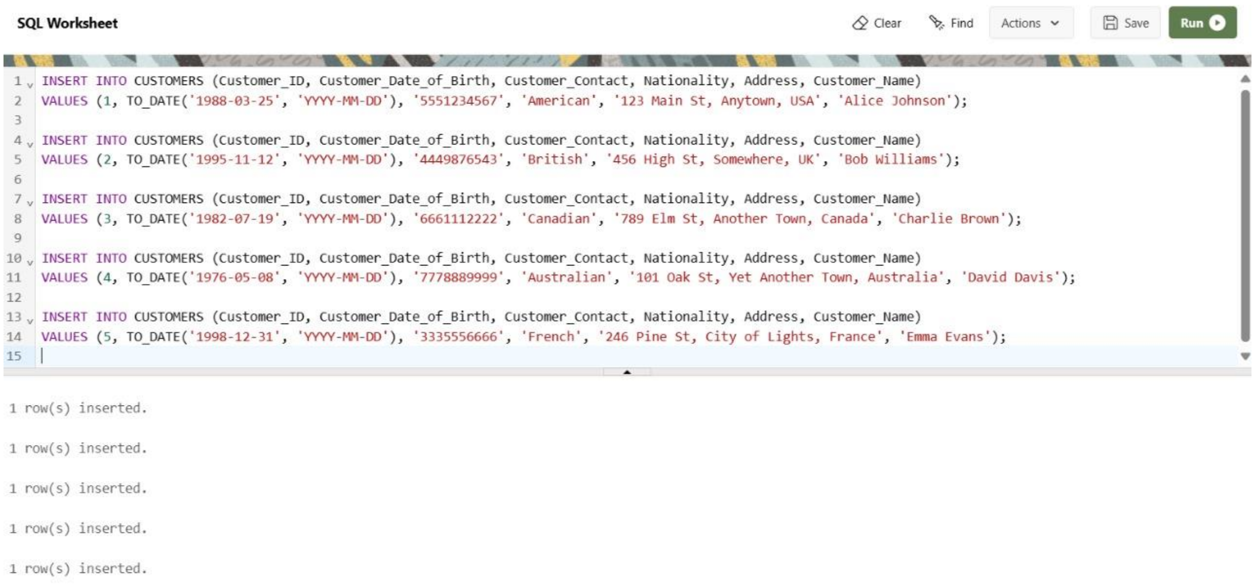


```sql
SELECT *
FROM EMPLOYEE;
```

EMPLOYEE_ID	EMPLOYEE_SALARY	EMPLOYEE_DATE_OF_BIRTH	EMPLOYEE_CONTACT_NUMBER	EMPLOYEE_FULL_NAME	EMPLOYEE_SEX	EMPLOYEE_EMAIL
1	50000	10-MAY-90	1234567890	John Doe	M	john@example.com
2	60000	15-AUG-85	9876543210	Jane Smith	F	jane@example.com
3	55000	28-FEB-93	5556667777	Alice Johnson	F	alice@example.com
4	62000	20-NOV-80	9998887777	Bob Williams	M	bob@example.com
5	58000	03-SEP-92	4443332222	Michael Brown	M	michael@example.com

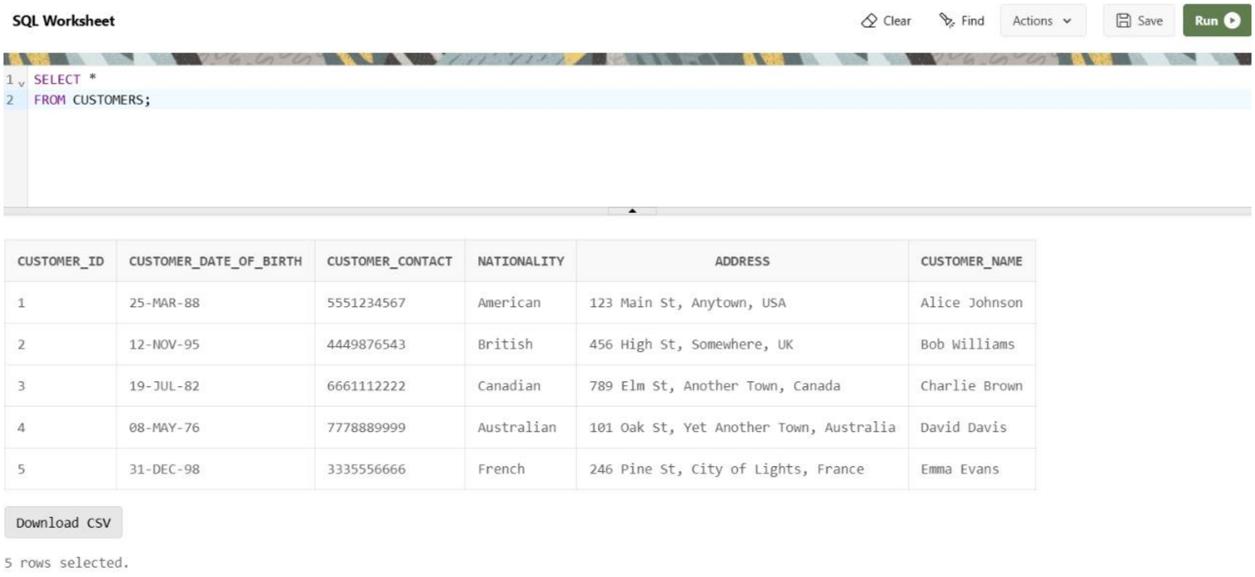
```
5 rows selected.
```

```
INSERT INTO CUSTOMERS (Customer_ID, Customer_Date_of_Birth,
Customer_Contact, Nationality, Address, Customer_Name)
VALUES (1, TO_DATE('1988-03-25', 'YYYY-MM-DD'), '5551234567', 'American', '123
Main St, Anytown, USA', 'Alice Johnson');

INSERT INTO CUSTOMERS (Customer_ID, Customer_Date_of_Birth,
Customer_Contact, Nationality, Address, Customer_Name)
VALUES (2, TO_DATE('1995-11-12', 'YYYY-MM-DD'), '4449876543', 'British', '456 High
St, Somewhere, UK', 'Bob Williams');

INSERT INTO CUSTOMERS (Customer_ID, Customer_Date_of_Birth,
Customer_Contact, Nationality, Address, Customer_Name)
VALUES (3, TO_DATE('1982-07-19', 'YYYY-MM-DD'), '6661112222', 'Canadian', '789
Elm St, Another Town, Canada', 'Charlie Brown');

INSERT INTO CUSTOMERS (Customer_ID, Customer_Date_of_Birth,
Customer_Contact, Nationality, Address, Customer_Name)
VALUES (4, TO_DATE('1976-05-08', 'YYYY-MM-DD'), '7778889999', 'Australian', '101
Oak St, Yet Another Town, Australia', 'David Davis');

INSERT INTO CUSTOMERS (Customer_ID, Customer_Date_of_Birth,
Customer_Contact, Nationality, Address, Customer_Name)
VALUES (5, TO_DATE('1998-12-31', 'YYYY-MM-DD'), '3335556666', 'French', '246 Pine
St, City of Lights, France', 'Emma Evans');
```



```sql
INSERT INTO CUSTOMERS (Customer_ID, Customer_Date_of_Birth, Customer_Contact, Nationality, Address, Customer_Name)
VALUES (1, TO_DATE('1988-03-25', 'YYYY-MM-DD'), '5551234567', 'American', '123 Main St, Anytown, USA', 'Alice Johnson');

INSERT INTO CUSTOMERS (Customer_ID, Customer_Date_of_Birth, Customer_Contact, Nationality, Address, Customer_Name)
VALUES (2, TO_DATE('1995-11-12', 'YYYY-MM-DD'), '4449876543', 'British', '456 High St, Somewhere, UK', 'Bob Williams');

INSERT INTO CUSTOMERS (Customer_ID, Customer_Date_of_Birth, Customer_Contact, Nationality, Address, Customer_Name)
VALUES (3, TO_DATE('1982-07-19', 'YYYY-MM-DD'), '6661112222', 'Canadian', '789 Elm St, Another Town, Canada', 'Charlie Brown');

INSERT INTO CUSTOMERS (Customer_ID, Customer_Date_of_Birth, Customer_Contact, Nationality, Address, Customer_Name)
VALUES (4, TO_DATE('1976-05-08', 'YYYY-MM-DD'), '7778889999', 'Australian', '101 Oak St, Yet Another Town, Australia', 'David Davis');

INSERT INTO CUSTOMERS (Customer_ID, Customer_Date_of_Birth, Customer_Contact, Nationality, Address, Customer_Name)
VALUES (5, TO_DATE('1998-12-31', 'YYYY-MM-DD'), '3335556666', 'French', '246 Pine St, City of Lights, France', 'Emma Evans');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

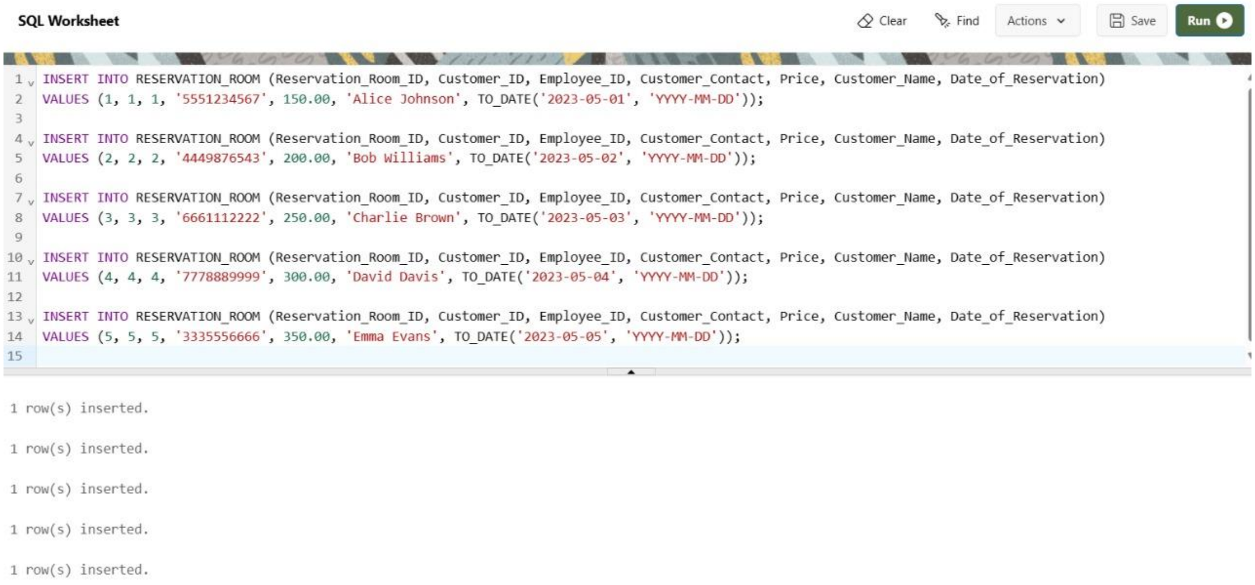


```sql
SELECT *
FROM CUSTOMERS;
```

CUSTOMER_ID	CUSTOMER_DATE_OF_BIRTH	CUSTOMER_CONTACT	NATIONALITY	ADDRESS	CUSTOMER_NAME
1	25-MAR-88	5551234567	American	123 Main St, Anytown, USA	Alice Johnson
2	12-NOV-95	4449876543	British	456 High St, Somewhere, UK	Bob Williams
3	19-JUL-82	6661112222	Canadian	789 Elm St, Another Town, Canada	Charlie Brown
4	08-MAY-76	7778889999	Australian	101 Oak St, Yet Another Town, Australia	David Davis
5	31-DEC-98	3335556666	French	246 Pine St, City of Lights, France	Emma Evans

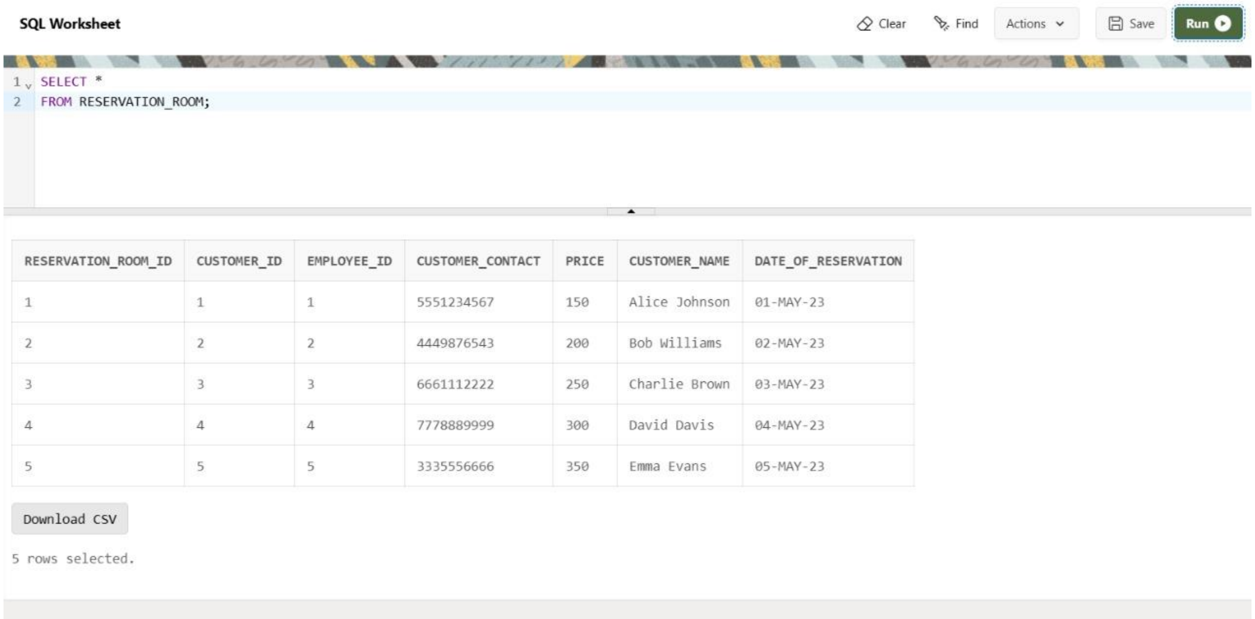
```
5 rows selected.
```

```
INSERT INTO RESERVATION_ROOM (Reservation_Room_ID, Customer_ID,
Employee_ID, Customer_Contact, Price, Customer_Name, Date_of_Reservation)
VALUES (1, 1, 1, '5551234567', 150.00, 'Alice Johnson', TO_DATE('2023-05-01',
'YYYY-MM-DD'));

INSERT INTO RESERVATION_ROOM (Reservation_Room_ID, Customer_ID,
Employee_ID, Customer_Contact, Price, Customer_Name, Date_of_Reservation)
VALUES (2, 2, 2, '4449876543', 200.00, 'Bob Williams', TO_DATE('2023-05-02', 'YYYY-
MM-DD'));

INSERT INTO RESERVATION_ROOM (Reservation_Room_ID, Customer_ID,
Employee_ID, Customer_Contact, Price, Customer_Name, Date_of_Reservation)
VALUES (3, 3, 3, '6661112222', 250.00, 'Charlie Brown', TO_DATE('2023-05-03',
'YYYY-MM-DD'));

INSERT INTO RESERVATION_ROOM (Reservation_Room_ID, Customer_ID,
Employee_ID, Customer_Contact, Price, Customer_Name, Date_of_Reservation)
VALUES (4, 4, 4, '7778889999', 300.00, 'David Davis', TO_DATE('2023-05-04', 'YYYY-
MM-DD'));

INSERT INTO RESERVATION_ROOM (Reservation_Room_ID, Customer_ID,
Employee_ID, Customer_Contact, Price, Customer_Name, Date_of_Reservation)
VALUES (5, 5, 5, '3335556666', 350.00, 'Emma Evans', TO_DATE('2023-05-05',
'YYYY-MM-DD'));
```



```sql
INSERT INTO RESERVATION_ROOM (Reservation_Room_ID, Customer_ID, Employee_ID, Customer_Contact, Price, Customer_Name, Date_of_Reservation)
VALUES (1, 1, 1, '5551234567', 150.00, 'Alice Johnson', TO_DATE('2023-05-01', 'YYYY-MM-DD'));

INSERT INTO RESERVATION_ROOM (Reservation_Room_ID, Customer_ID, Employee_ID, Customer_Contact, Price, Customer_Name, Date_of_Reservation)
VALUES (2, 2, 2, '4449876543', 200.00, 'Bob Williams', TO_DATE('2023-05-02', 'YYYY-MM-DD'));

INSERT INTO RESERVATION_ROOM (Reservation_Room_ID, Customer_ID, Employee_ID, Customer_Contact, Price, Customer_Name, Date_of_Reservation)
VALUES (3, 3, 3, '6661112222', 250.00, 'Charlie Brown', TO_DATE('2023-05-03', 'YYYY-MM-DD'));

INSERT INTO RESERVATION_ROOM (Reservation_Room_ID, Customer_ID, Employee_ID, Customer_Contact, Price, Customer_Name, Date_of_Reservation)
VALUES (4, 4, 4, '7778889999', 300.00, 'David Davis', TO_DATE('2023-05-04', 'YYYY-MM-DD'));

INSERT INTO RESERVATION_ROOM (Reservation_Room_ID, Customer_ID, Employee_ID, Customer_Contact, Price, Customer_Name, Date_of_Reservation)
VALUES (5, 5, 5, '3335556666', 350.00, 'Emma Evans', TO_DATE('2023-05-05', 'YYYY-MM-DD'));
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

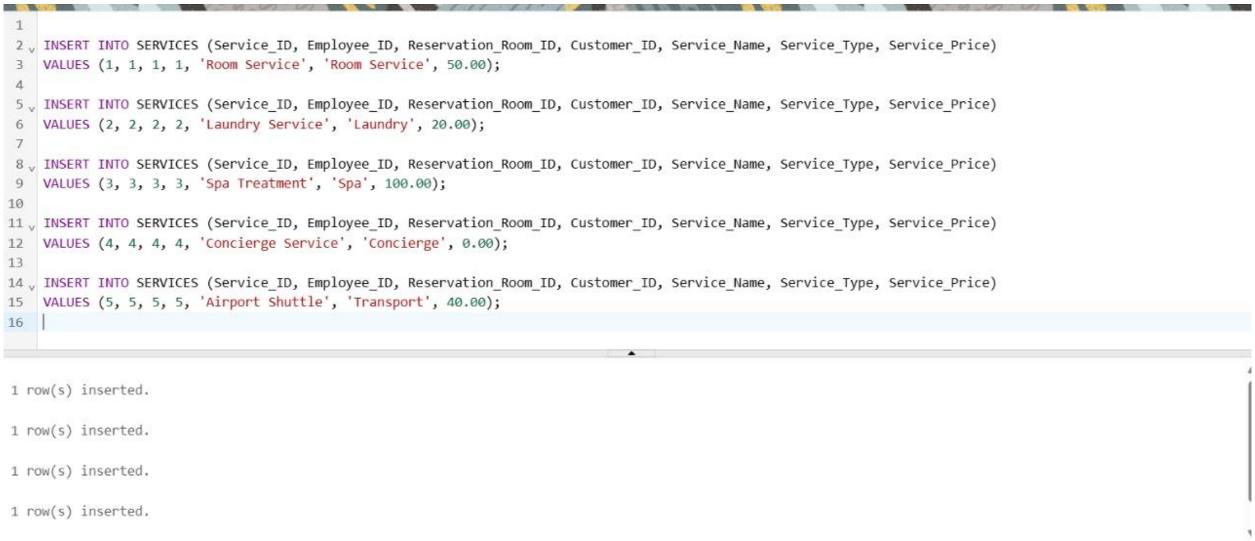


```sql
SELECT *
FROM RESERVATION_ROOM;
```

RESERVATION_ROOM_ID	CUSTOMER_ID	EMPLOYEE_ID	CUSTOMER_CONTACT	PRICE	CUSTOMER_NAME	DATE_OF_RESERVATION
1	1	1	5551234567	150	Alice Johnson	01-MAY-23
2	2	2	4449876543	200	Bob Williams	02-MAY-23
3	3	3	6661112222	250	Charlie Brown	03-MAY-23
4	4	4	7778889999	300	David Davis	04-MAY-23
5	5	5	3335556666	350	Emma Evans	05-MAY-23

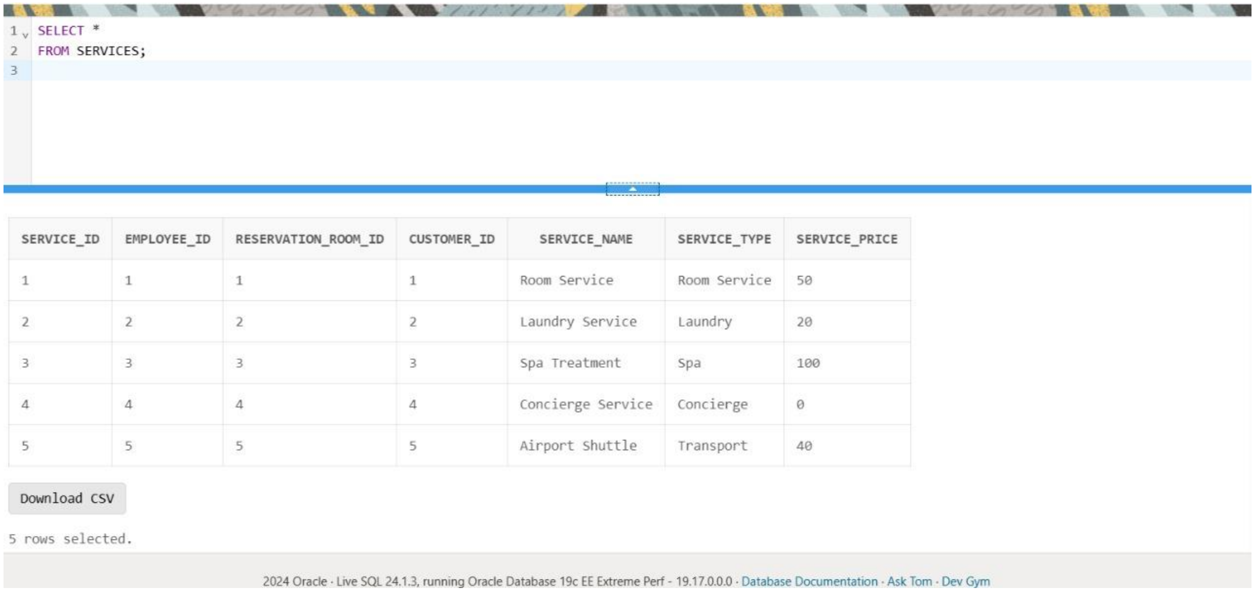
```
5 rows selected.
```

```
INSERT INTO SERVICES (Service_ID, Employee_ID, Reservation_Room_ID,
Customer_ID, Service_Name, Service_Type, Service_Price)
VALUES (1, 1, 1, 1, 'Room Service', 'Room Service', 50.00);

INSERT INTO SERVICES (Service_ID, Employee_ID, Reservation_Room_ID,
Customer_ID, Service_Name, Service_Type, Service_Price)
VALUES (2, 2, 2, 2, 'Laundry Service', 'Laundry', 20.00);

INSERT INTO SERVICES (Service_ID, Employee_ID, Reservation_Room_ID,
Customer_ID, Service_Name, Service_Type, Service_Price)
VALUES (3, 3, 3, 3, 'Spa Treatment', 'Spa', 100.00);

INSERT INTO SERVICES (Service_ID, Employee_ID, Reservation_Room_ID,
Customer_ID, Service_Name, Service_Type, Service_Price)
VALUES (4, 4, 4, 4, 'Concierge Service', 'Concierge', 0.00);

INSERT INTO SERVICES (Service_ID, Employee_ID, Reservation_Room_ID,
Customer_ID, Service_Name, Service_Type, Service_Price)
VALUES (5, 5, 5, 5, 'Airport Shuttle', 'Transport', 40.00);
```



```
INSERT INTO SERVICES (Service_ID, Employee_ID, Reservation_Room_ID, Customer_ID, Service_Name, Service_Type, Service_Price)
VALUES (1, 1, 1, 1, 'Room Service', 'Room Service', 50.00);

INSERT INTO SERVICES (Service_ID, Employee_ID, Reservation_Room_ID, Customer_ID, Service_Name, Service_Type, Service_Price)
VALUES (2, 2, 2, 2, 'Laundry Service', 'Laundry', 20.00);

INSERT INTO SERVICES (Service_ID, Employee_ID, Reservation_Room_ID, Customer_ID, Service_Name, Service_Type, Service_Price)
VALUES (3, 3, 3, 3, 'Spa Treatment', 'Spa', 100.00);

INSERT INTO SERVICES (Service_ID, Employee_ID, Reservation_Room_ID, Customer_ID, Service_Name, Service_Type, Service_Price)
VALUES (4, 4, 4, 4, 'Concierge Service', 'Concierge', 0.00);

INSERT INTO SERVICES (Service_ID, Employee_ID, Reservation_Room_ID, Customer_ID, Service_Name, Service_Type, Service_Price)
VALUES (5, 5, 5, 5, 'Airport Shuttle', 'Transport', 40.00);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```



```
SELECT *
FROM SERVICES;
```

SERVICE_ID	EMPLOYEE_ID	RESERVATION_ROOM_ID	CUSTOMER_ID	SERVICE_NAME	SERVICE_TYPE	SERVICE_PRICE
1	1	1	1	Room Service	Room Service	50
2	2	2	2	Laundry Service	Laundry	20
3	3	3	3	Spa Treatment	Spa	100
4	4	4	4	Concierge Service	Concierge	0
5	5	5	5	Airport Shuttle	Transport	40

```
5 rows selected.
```

2024 Oracle · Live SQL 24.1.3, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym

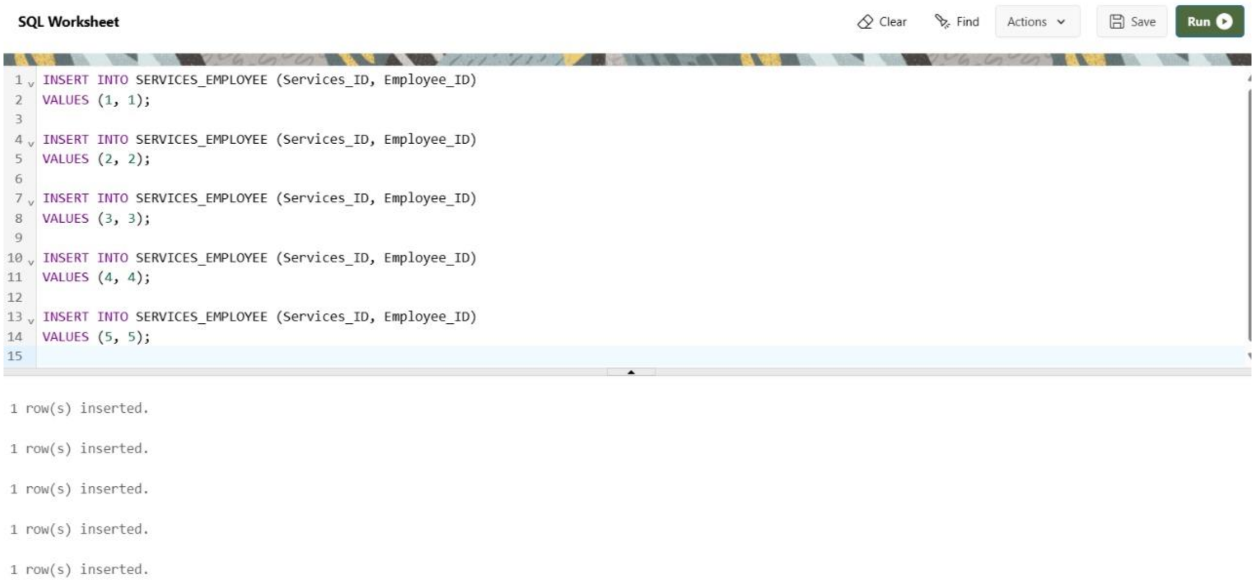
INSERT INTO SERVICES_EMPLOYEE (Services_ID, Employee_ID)
VALUES (1, 1);

INSERT INTO SERVICES_EMPLOYEE (Services_ID, Employee_ID)
VALUES (2, 2);

INSERT INTO SERVICES_EMPLOYEE (Services_ID, Employee_ID)
VALUES (3, 3);

INSERT INTO SERVICES_EMPLOYEE (Services_ID, Employee_ID)
VALUES (4, 4);

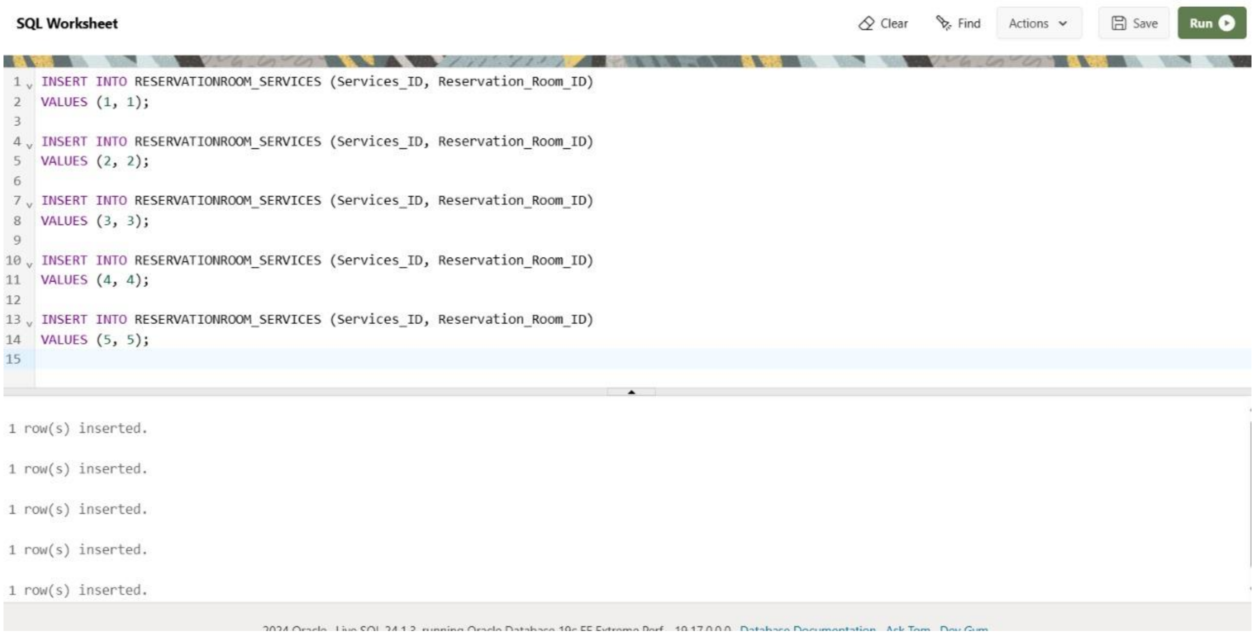
INSERT INTO SERVICES_EMPLOYEE (Services_ID, Employee_ID)
VALUES (5, 5);



```
INSERT INTO SERVICES_EMPLOYEE (Services_ID, Employee_ID)
VALUES (1, 1);

INSERT INTO SERVICES_EMPLOYEE (Services_ID, Employee_ID)
VALUES (2, 2);

INSERT INTO SERVICES_EMPLOYEE (Services_ID, Employee_ID)
VALUES (3, 3);

INSERT INTO SERVICES_EMPLOYEE (Services_ID, Employee_ID)
VALUES (4, 4);

INSERT INTO SERVICES_EMPLOYEE (Services_ID, Employee_ID)
VALUES (5, 5);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

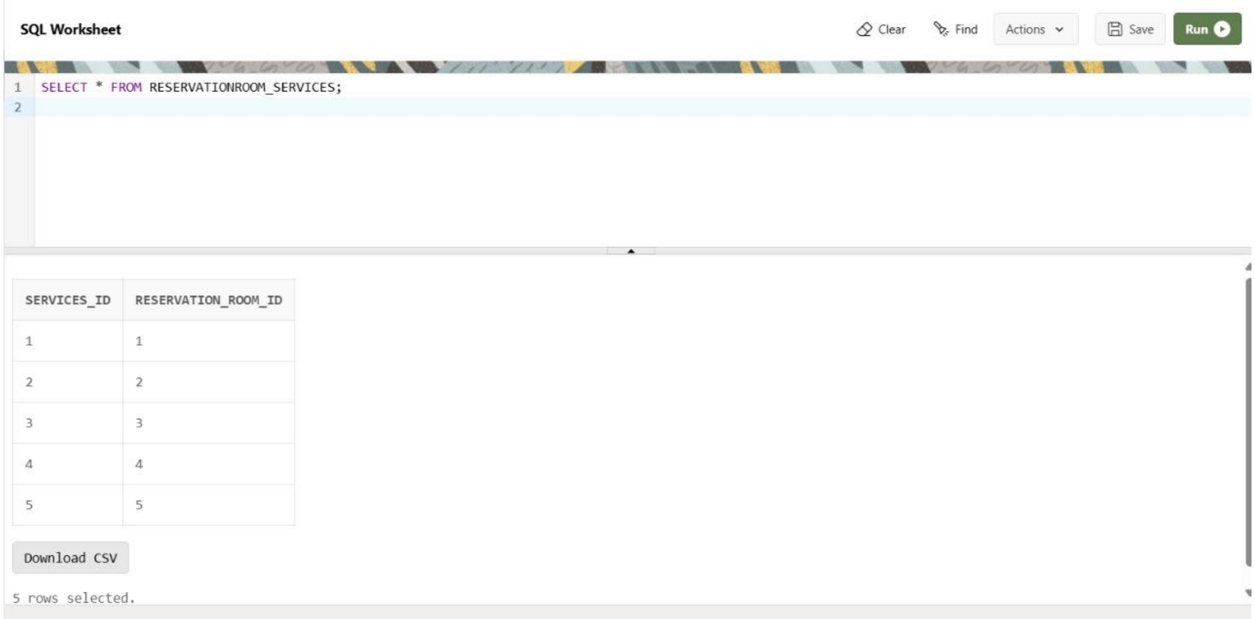
```
INSERT INTO RESERVATIONROOM_SERVICES (Services_ID,
Reservation_Room_ID)
VALUES (1, 1);

INSERT INTO RESERVATIONROOM_SERVICES (Services_ID,
Reservation_Room_ID)
VALUES (2, 2);

INSERT INTO RESERVATIONROOM_SERVICES (Services_ID,
Reservation_Room_ID)
VALUES (3, 3);

INSERT INTO RESERVATIONROOM_SERVICES (Services_ID,
Reservation_Room_ID)
VALUES (4, 4);

INSERT INTO RESERVATIONROOM_SERVICES (Services_ID,
Reservation_Room_ID)
VALUES (5, 5);
```



```
INSERT INTO RESERVATIONROOM_SERVICES (Services_ID, Reservation_Room_ID)
VALUES (1, 1);

INSERT INTO RESERVATIONROOM_SERVICES (Services_ID, Reservation_Room_ID)
VALUES (2, 2);

INSERT INTO RESERVATIONROOM_SERVICES (Services_ID, Reservation_Room_ID)
VALUES (3, 3);

INSERT INTO RESERVATIONROOM_SERVICES (Services_ID, Reservation_Room_ID)
VALUES (4, 4);

INSERT INTO RESERVATIONROOM_SERVICES (Services_ID, Reservation_Room_ID)
VALUES (5, 5);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

2024 Oracle · Live SQL 24.1.3, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym

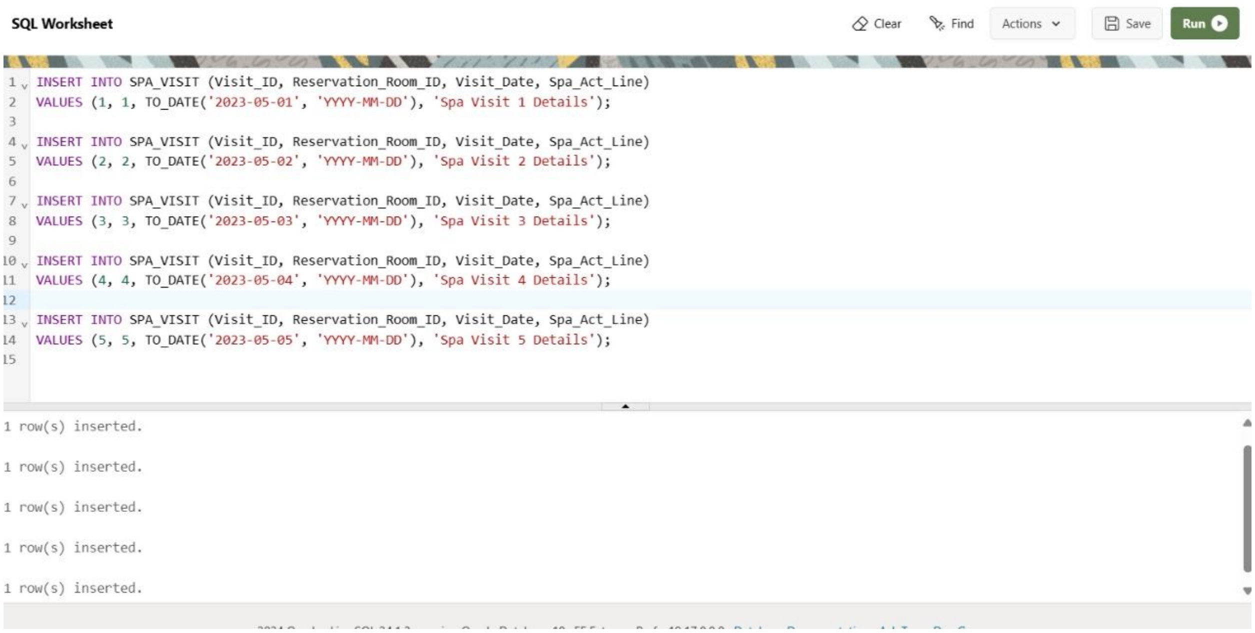

SQL Worksheet
Clear
Find
Actions
Save
Run

```
SELECT * FROM RESERVATIONROOM_SERVICES;
```

SERVICES_ID	RESERVATION_ROOM_ID
1	1
2	2
3	3
4	4
5	5

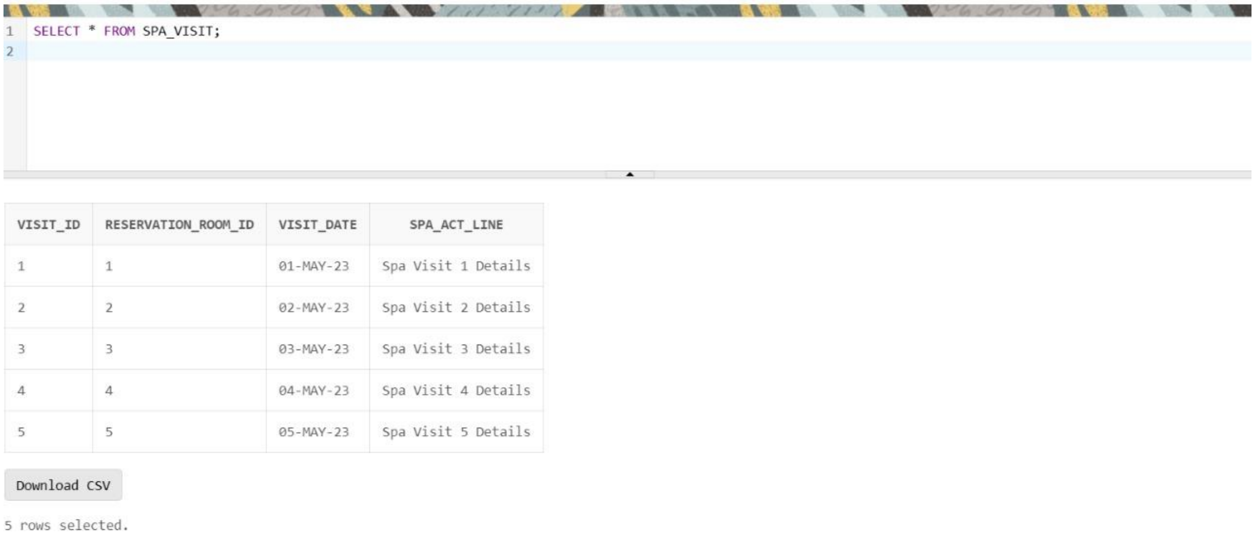
```
5 rows selected.
```

```
INSERT INTO SPA_VISIT (Visit_ID, Reservation_Room_ID, Visit_Date, Spa_Act_Line)
VALUES (1, 1, TO_DATE('2023-05-01', 'YYYY-MM-DD'), 'Spa Visit 1 Details');

INSERT INTO SPA_VISIT (Visit_ID, Reservation_Room_ID, Visit_Date, Spa_Act_Line)
VALUES (2, 2, TO_DATE('2023-05-02', 'YYYY-MM-DD'), 'Spa Visit 2 Details');

INSERT INTO SPA_VISIT (Visit_ID, Reservation_Room_ID, Visit_Date, Spa_Act_Line)
VALUES (3, 3, TO_DATE('2023-05-03', 'YYYY-MM-DD'), 'Spa Visit 3 Details');

INSERT INTO SPA_VISIT (Visit_ID, Reservation_Room_ID, Visit_Date, Spa_Act_Line)
VALUES (4, 4, TO_DATE('2023-05-04', 'YYYY-MM-DD'), 'Spa Visit 4 Details');

INSERT INTO SPA_VISIT (Visit_ID, Reservation_Room_ID, Visit_Date, Spa_Act_Line)
VALUES (5, 5, TO_DATE('2023-05-05', 'YYYY-MM-DD'), 'Spa Visit 5 Details');
```



```
INSERT INTO SPA_VISIT (Visit_ID, Reservation_Room_ID, Visit_Date, Spa_Act_Line)
VALUES (1, 1, TO_DATE('2023-05-01', 'YYYY-MM-DD'), 'Spa Visit 1 Details');

INSERT INTO SPA_VISIT (Visit_ID, Reservation_Room_ID, Visit_Date, Spa_Act_Line)
VALUES (2, 2, TO_DATE('2023-05-02', 'YYYY-MM-DD'), 'Spa Visit 2 Details');

INSERT INTO SPA_VISIT (Visit_ID, Reservation_Room_ID, Visit_Date, Spa_Act_Line)
VALUES (3, 3, TO_DATE('2023-05-03', 'YYYY-MM-DD'), 'Spa Visit 3 Details');

INSERT INTO SPA_VISIT (Visit_ID, Reservation_Room_ID, Visit_Date, Spa_Act_Line)
VALUES (4, 4, TO_DATE('2023-05-04', 'YYYY-MM-DD'), 'Spa Visit 4 Details');

INSERT INTO SPA_VISIT (Visit_ID, Reservation_Room_ID, Visit_Date, Spa_Act_Line)
VALUES (5, 5, TO_DATE('2023-05-05', 'YYYY-MM-DD'), 'Spa Visit 5 Details');
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```



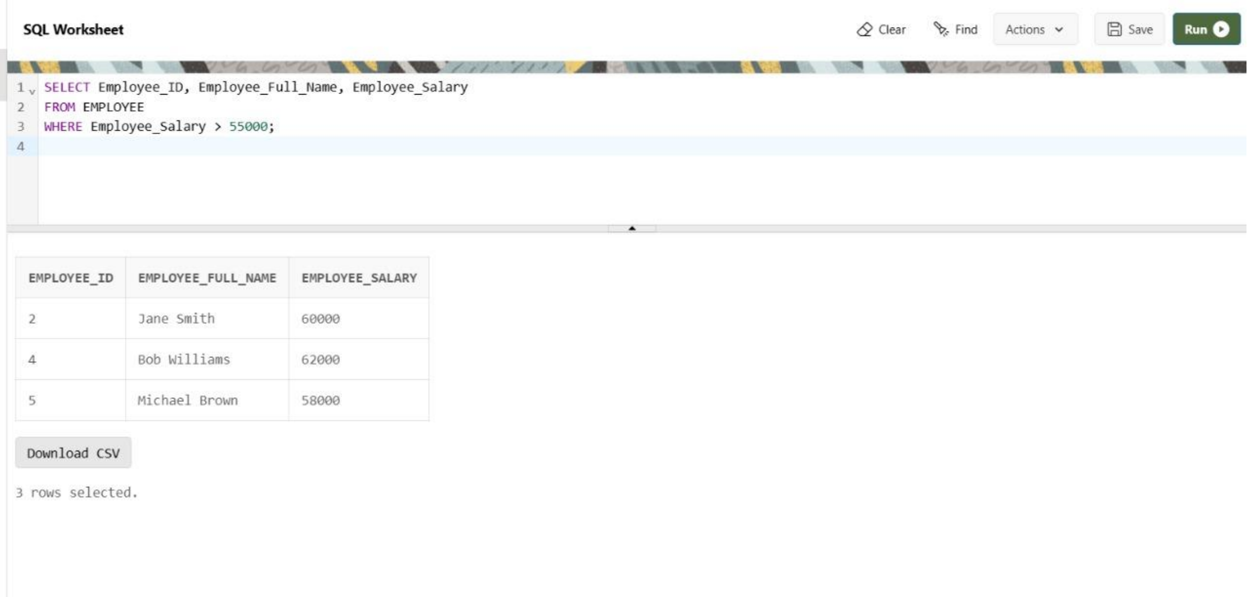
```
SELECT * FROM SPA_VISIT;
```

VISIT_ID	RESERVATION_ROOM_ID	VISIT_DATE	SPA_ACT_LINE
1	1	01-MAY-23	Spa Visit 1 Details
2	2	02-MAY-23	Spa Visit 2 Details
3	3	03-MAY-23	Spa Visit 3 Details
4	4	04-MAY-23	Spa Visit 4 Details
5	5	05-MAY-23	Spa Visit 5 Details

```
5 rows selected.
```

- QUERY

```
SELECT Employee_ID, Employee_Full_Name, Employee_Salary
FROM EMPLOYEE
WHERE Employee_Salary > 55000;
```

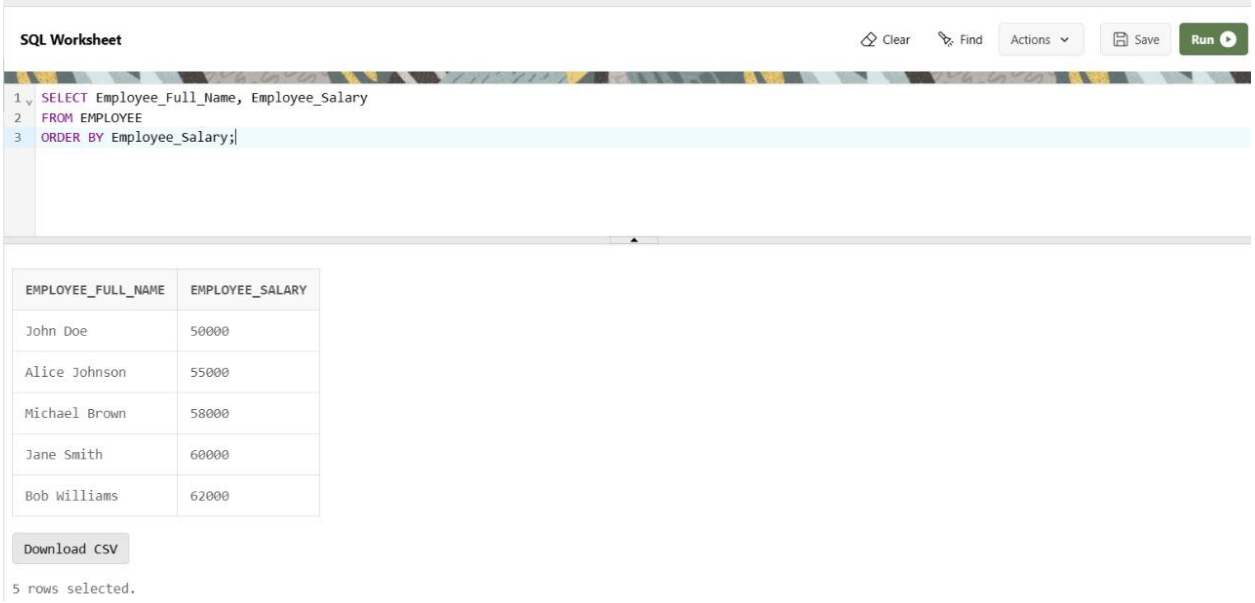


```
SELECT Employee_ID, Employee_Full_Name, Employee_Salary
FROM EMPLOYEE
WHERE Employee_Salary > 55000;
```

EMPLOYEE_ID	EMPLOYEE_FULL_NAME	EMPLOYEE_SALARY
2	Jane Smith	60000
4	Bob Williams	62000
5	Michael Brown	58000

```
3 rows selected.
```

```
SELECT Employee_Full_Name, Employee_Salary
FROM EMPLOYEE
ORDER BY Employee_Salary;
```

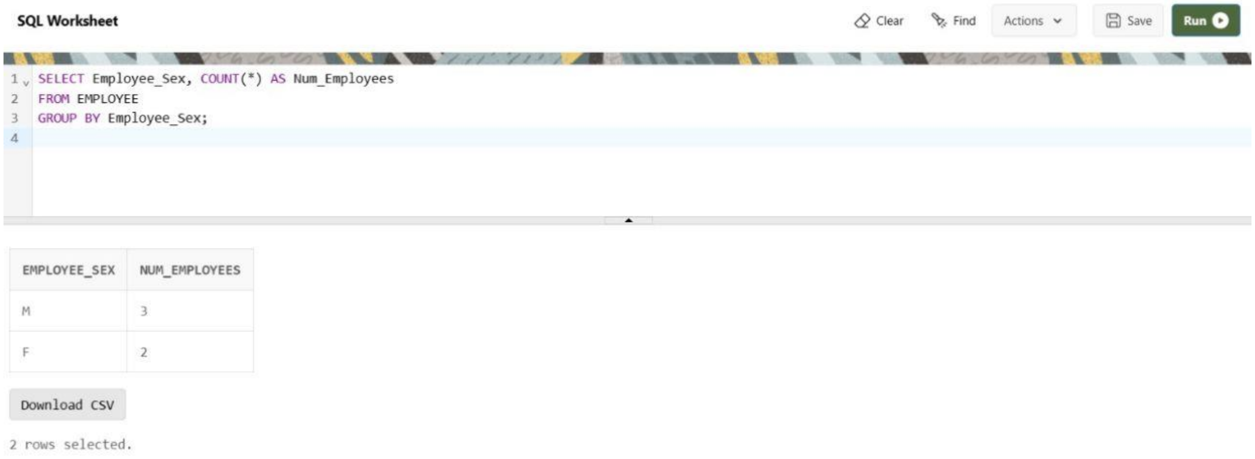


```
SELECT Employee_Full_Name, Employee_Salary
FROM EMPLOYEE
ORDER BY Employee_Salary;
```

EMPLOYEE_FULL_NAME	EMPLOYEE_SALARY
John Doe	50000
Alice Johnson	55000
Michael Brown	58000
Jane Smith	60000
Bob Williams	62000

```
5 rows selected.
```

```
SELECT Employee_Sex, COUNT(*) AS Num_Employees
FROM EMPLOYEE
GROUP BY Employee_Sex;
```

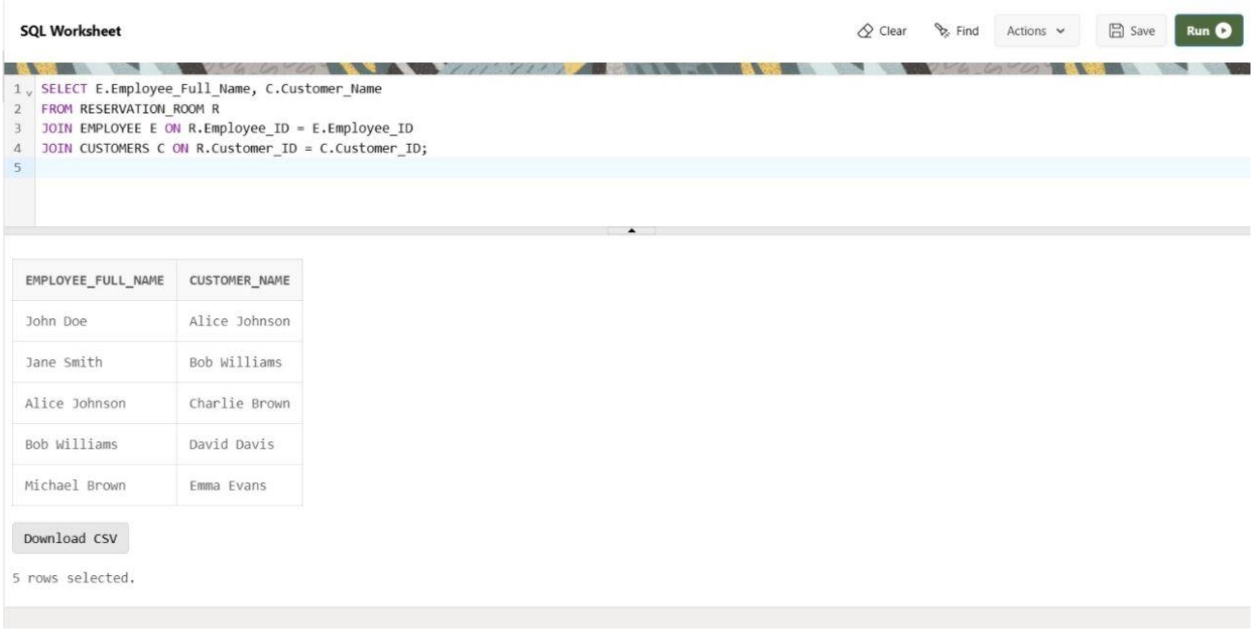


```
SELECT Employee_Sex, COUNT(*) AS Num_Employees
FROM EMPLOYEE
GROUP BY Employee_Sex;
```

EMPLOYEE_SEX	NUM_EMPLOYEES
M	3
F	2

```
2 rows selected.
```

```
SELECT E.Employee_Full_Name, C.Customer_Name
FROM RESERVATION_ROOM R
JOIN EMPLOYEE E ON R.Employee_ID = E.Employee_ID
JOIN CUSTOMERS C ON R.Customer_ID = C.Customer_ID;
```



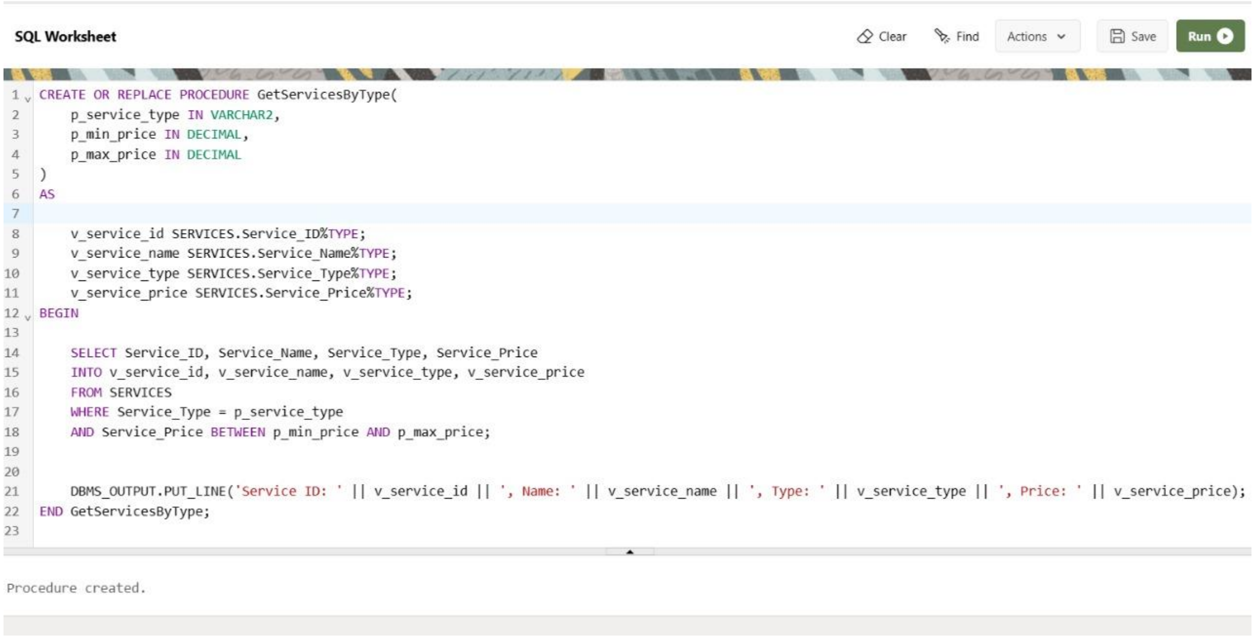
```
SELECT E.Employee_Full_Name, C.Customer_Name
FROM RESERVATION_ROOM R
JOIN EMPLOYEE E ON R.Employee_ID = E.Employee_ID
JOIN CUSTOMERS C ON R.Customer_ID = C.Customer_ID;
```

EMPLOYEE_FULL_NAME	CUSTOMER_NAME
John Doe	Alice Johnson
Jane Smith	Bob Williams
Alice Johnson	Charlie Brown
Bob Williams	David Davis
Michael Brown	Emma Evans

```
5 rows selected.
```

- PROCEDURE:

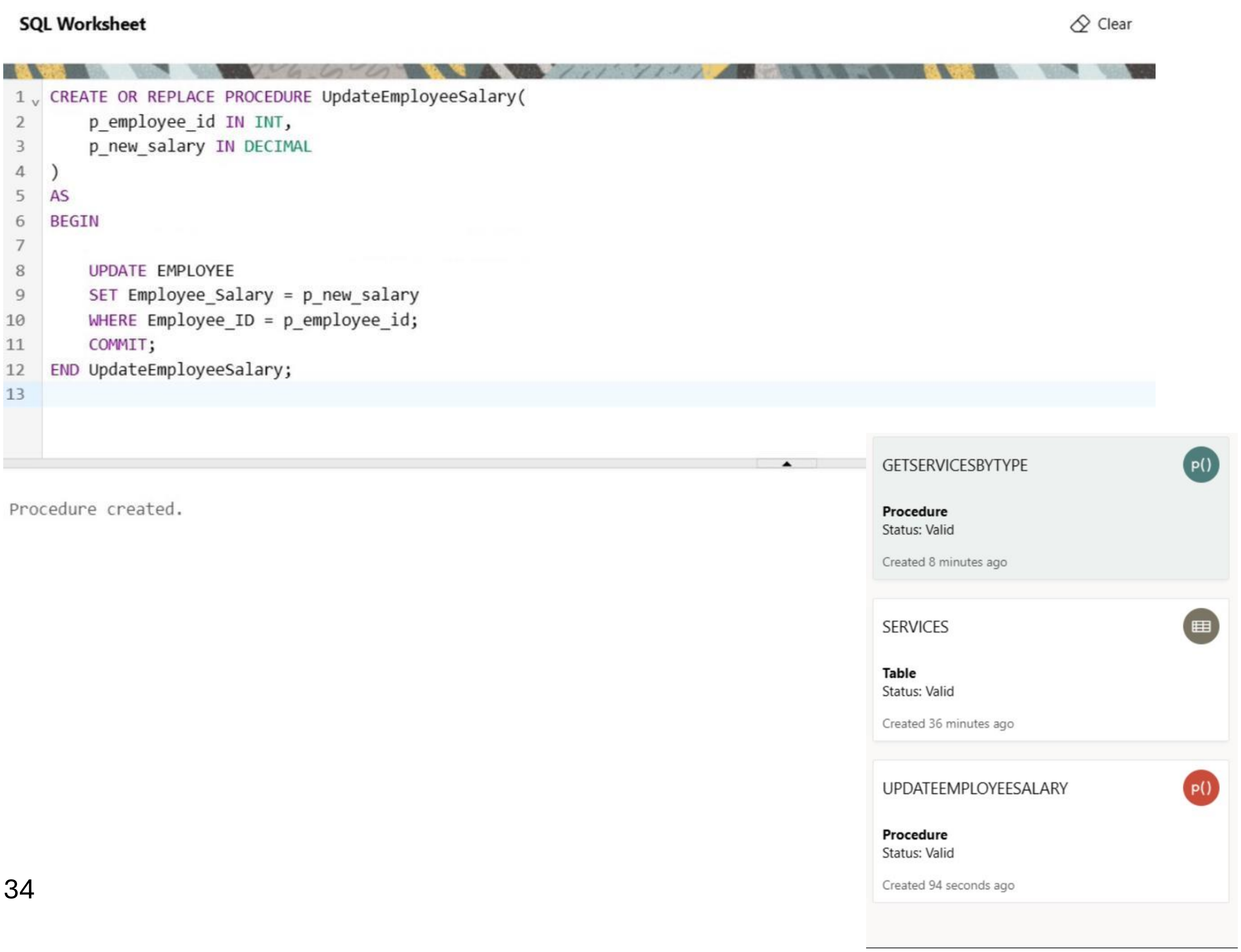
```
REATE OR REPLACE PROCEDURE GetServicesByType(
p_service_type IN VARCHAR2,
p_min_price IN DECIMAL,
p_max_price IN DECIMAL
)
AS

v_service_id SERVICES.Service_ID%TYPE;
v_service_name SERVICES.Service_Name%TYPE;
v_service_type SERVICES.Service_Type%TYPE;
v_service_price SERVICES.Service_Price%TYPE;
BEGIN

SELECT Service_ID, Service_Name, Service_Type, Service_Price
INTO v_service_id, v_service_name, v_service_type, v_service_price
FROM SERVICES
WHERE Service_Type = p_service_type
AND Service_Price BETWEEN p_min_price AND p_max_price;


DBMS_OUTPUT.PUT_LINE('Service ID: ' || v_service_id || ', Name: ' || v_service_name
|| ', Type: ' || v_service_type || ', Price: ' || v_service_price);
END GetServicesByType;
```



```
CREATE OR REPLACE PROCEDURE GetServicesByType(
    p_service_type IN VARCHAR2,
    p_min_price IN DECIMAL,
    p_max_price IN DECIMAL
)
AS

    v_service_id SERVICES.Service_ID%TYPE;
    v_service_name SERVICES.Service_Name%TYPE;
    v_service_type SERVICES.Service_Type%TYPE;
    v_service_price SERVICES.Service_Price%TYPE;
BEGIN

    SELECT Service_ID, Service_Name, Service_Type, Service_Price
    INTO v_service_id, v_service_name, v_service_type, v_service_price
    FROM SERVICES
    WHERE Service_Type = p_service_type
    AND Service_Price BETWEEN p_min_price AND p_max_price;


    DBMS_OUTPUT.PUT_LINE('Service ID: ' || v_service_id || ', Name: ' || v_service_name || ', Type: ' || v_service_type || ', Price: ' || v_service_price);
END GetServicesByType;

```

```
Procedure created.
```

```
CREATE OR REPLACE PROCEDURE UpdateEmployeeSalary(
p_employee_id IN INT,
p_new_salary IN DECIMAL
)
AS
BEGIN
-- Update employee salary based on employee ID
UPDATE EMPLOYEE
SET Employee_Salary = p_new_salary
WHERE Employee_ID = p_employee_id;
COMMIT;
END UpdateEmployeeSalary;
```



```
CREATE OR REPLACE PROCEDURE UpdateEmployeeSalary(
    p_employee_id IN INT,
    p_new_salary IN DECIMAL
)
AS
BEGIN

    UPDATE EMPLOYEE
    SET Employee_Salary = p_new_salary
    WHERE Employee_ID = p_employee_id;
    COMMIT;
END UpdateEmployeeSalary;

```

```
Procedure created.
```

GETSERVICESBYTYPE

Procedure
Status: Valid

Created 8 minutes ago

SERVICES

Table
Status: Valid

Created 36 minutes ago

UPDATEEMPLOYEESALARY

Procedure
Status: Valid

Created 94 seconds ago