i2i Systems
innovation to integration

i2i Academy

Training Document

Topic	Oracle SQL Language Fundamentals I
Document Name	SQL03-EX-01-05

Exercise SQL03-EX-01:

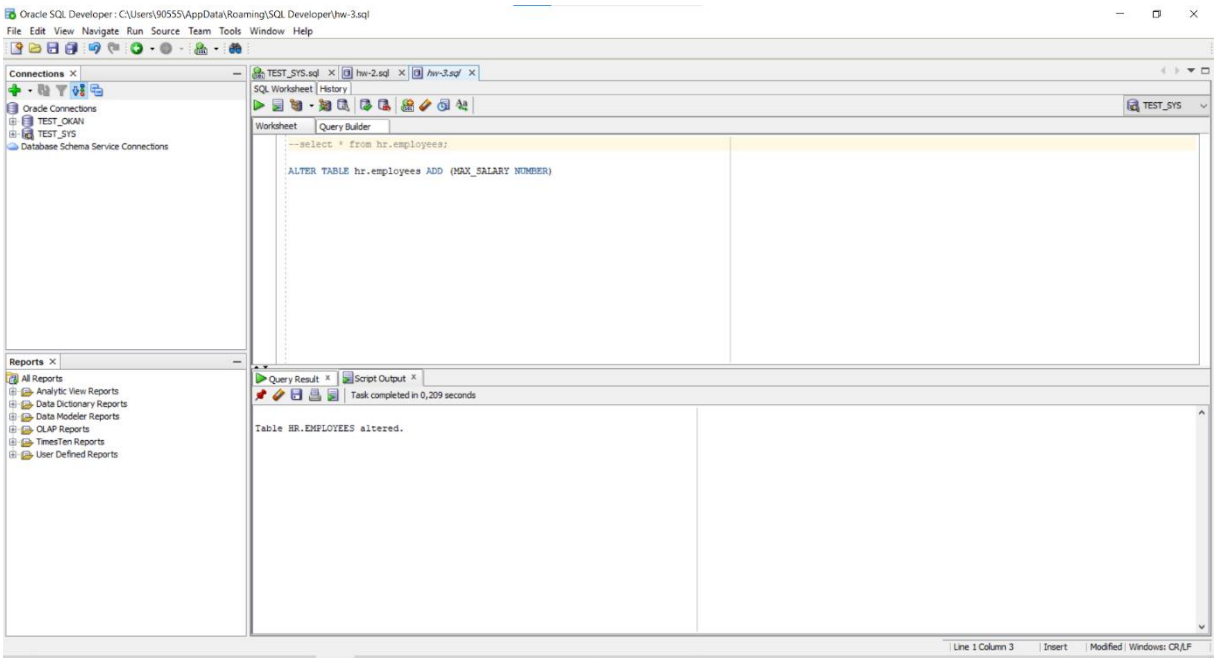
Definiton : Write followig SQL queries:

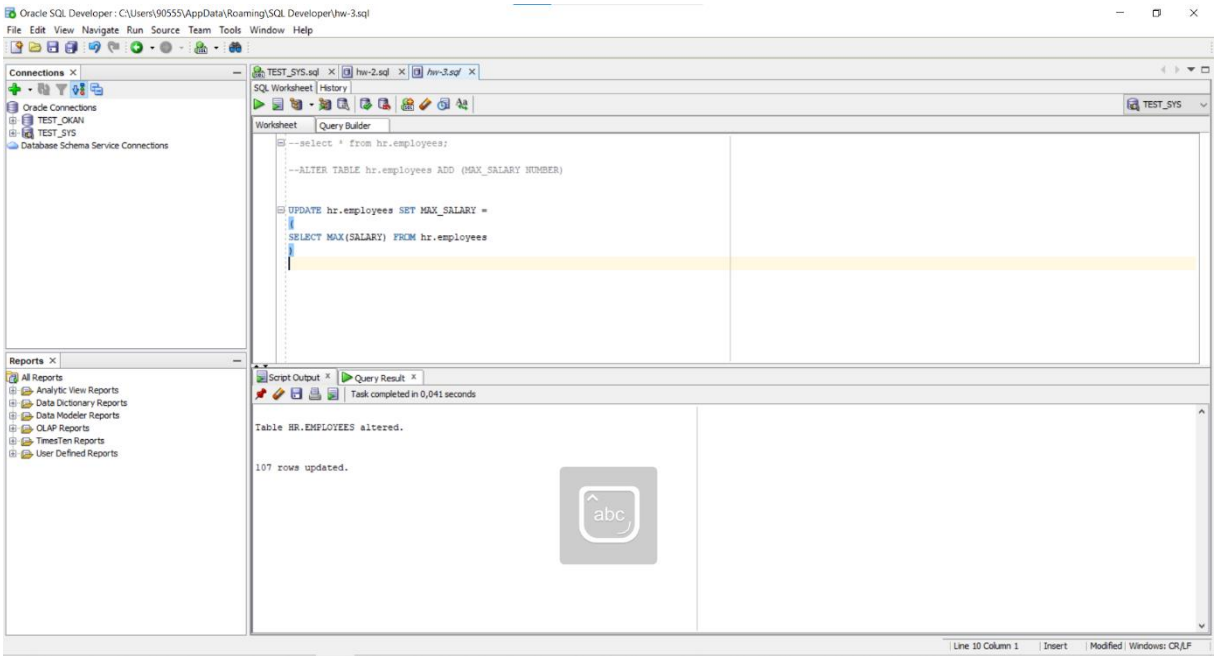
- Add a colum to employees table named MAX_SALARY.
- Update MAX_SALARY with maximum salary amount with subquery.
- Delete employee who have minimum salary using subquery.

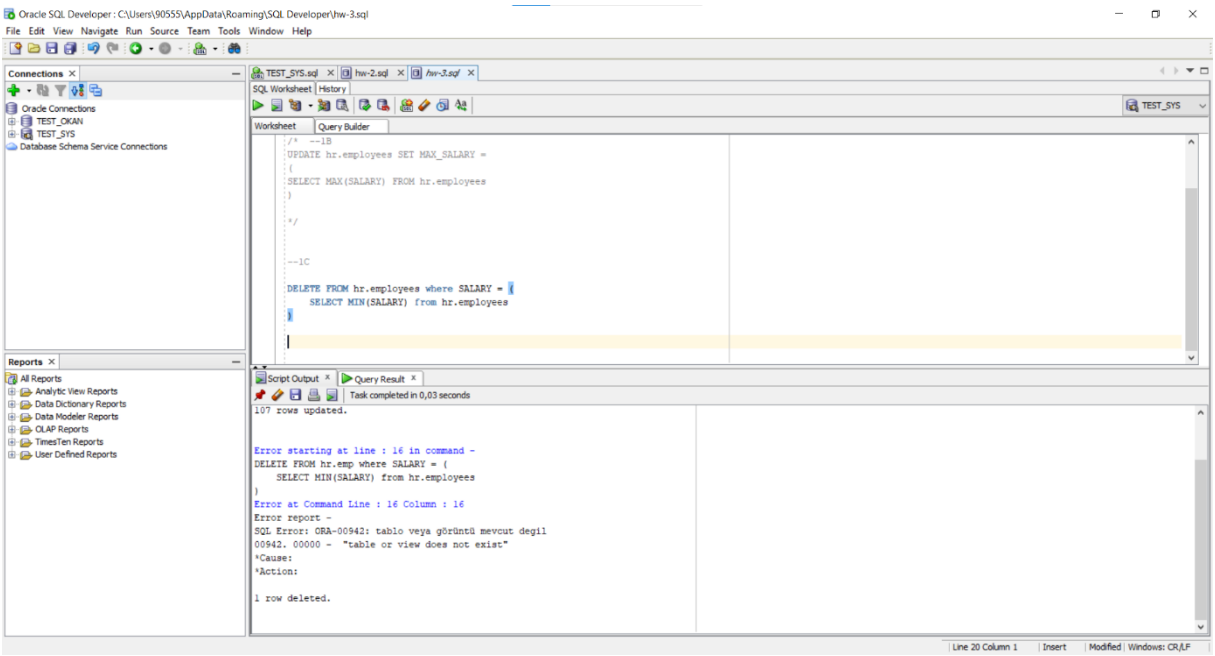
SQL:

```
--1A

        ALTER TABLE hr.employees ADD (MAX_SALARY NUMBER) --1A

--1B

        UPDATE hr.employees SET MAX_SALARY =

                (

                SELECT MAX(SALARY) FROM hr.employees

                )

--1C

        DELETE FROM hr.employees where SALARY = (

                SELECT MIN(SALARY) from hr.employees

        )
```

Screenshot:


Oracle SQL Developer : C:\Users\90555\AppData\Roaming\SQL Developer\hw-3.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
TEST_OKAN
TEST_SYS
Database Schema Service Connections
TEST_SYS.sql
hw-2.sql
hw-3.sql
SQL Worksheet
History
Worksheet
Query Builder
--select * from hr.employees;
ALTER TABLE hr.employees ADD (MAX_SALARY NUMBER)
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Query Result
Script Output
Task completed in 0,209 seconds
Table HR.EMPLOYEES altered.
Line 1 Column 3
Insert
Modified
Windows: CR/LF


Oracle SQL Developer : C:\Users\90555\AppData\Roaming\SQL Developer\hw-3.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
TEST_OKAN
TEST_SYS
Database Schema Service Connections
TEST_SYS.sql
hw-2.sql
hw-3.sql
SQL Worksheet
History
Worksheet
Query Builder
--select * from hr.employees;
--ALTER TABLE hr.employees ADD (MAX_SALARY NUMBER)
UPDATE hr.employees SET MAX_SALARY =
(
SELECT MAX(SALARY) FROM hr.employees
)
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Script Output
Query Result
Task completed in 0,041 seconds
Table HR.EMPLOYEES altered.
107 rows updated.
Line 10 Column 1
Insert
Modified
Windows: CR/LF



Exercise SQL03-EX-02:

Definiton : Write followig SQL queries:

- Define index (named DPR_NAME_IDX) on DEPARTMENT_NAME column of DEPARTMENTS table.
- Define constraint (named CNSTR_SALARY) on employee salary. (Salary must be between 1000$ and 100.000$)
- Drop defined index.
- Enable, disable, drop defined constraint.

SQL:

--2A

CREATE INDEX DPR_NAME_IDX ON hr.departments(DEPARTMENT_NAME)

--2B

ALTER TABLE hr.employees ADD CONSTRAINT CNSTR_SALARY CHECK(SALARY BETWEEN 1000 AND 100000);

--2C

DROP INDEX DPR_NAME_IDX

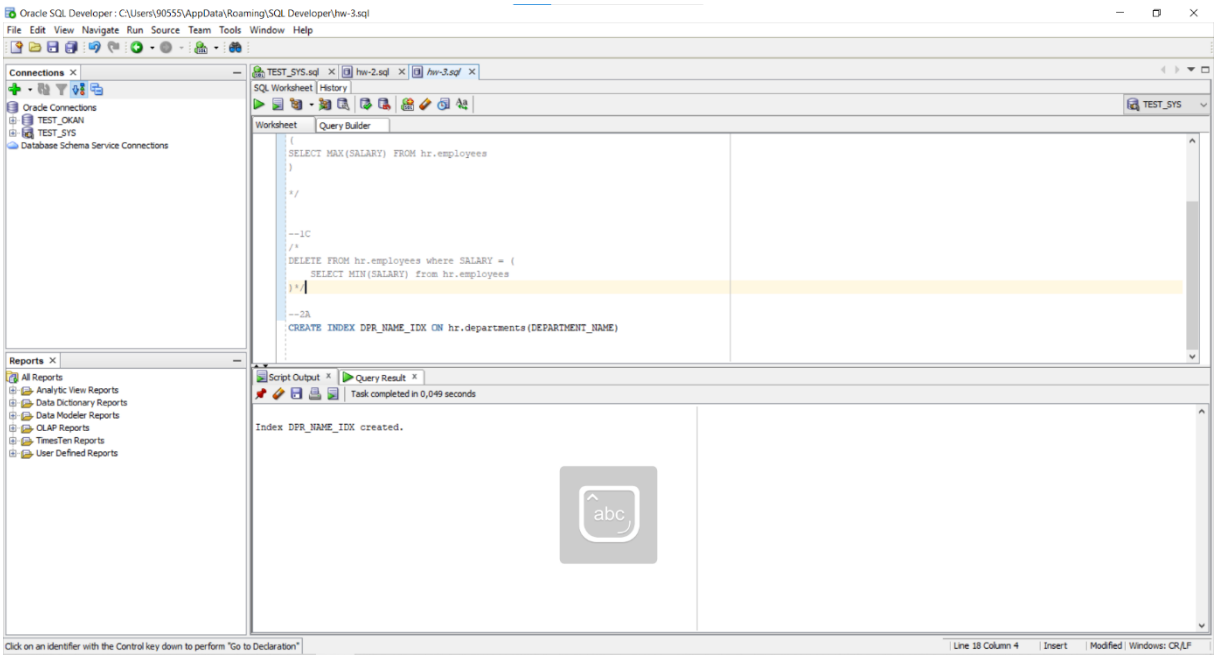
--2D

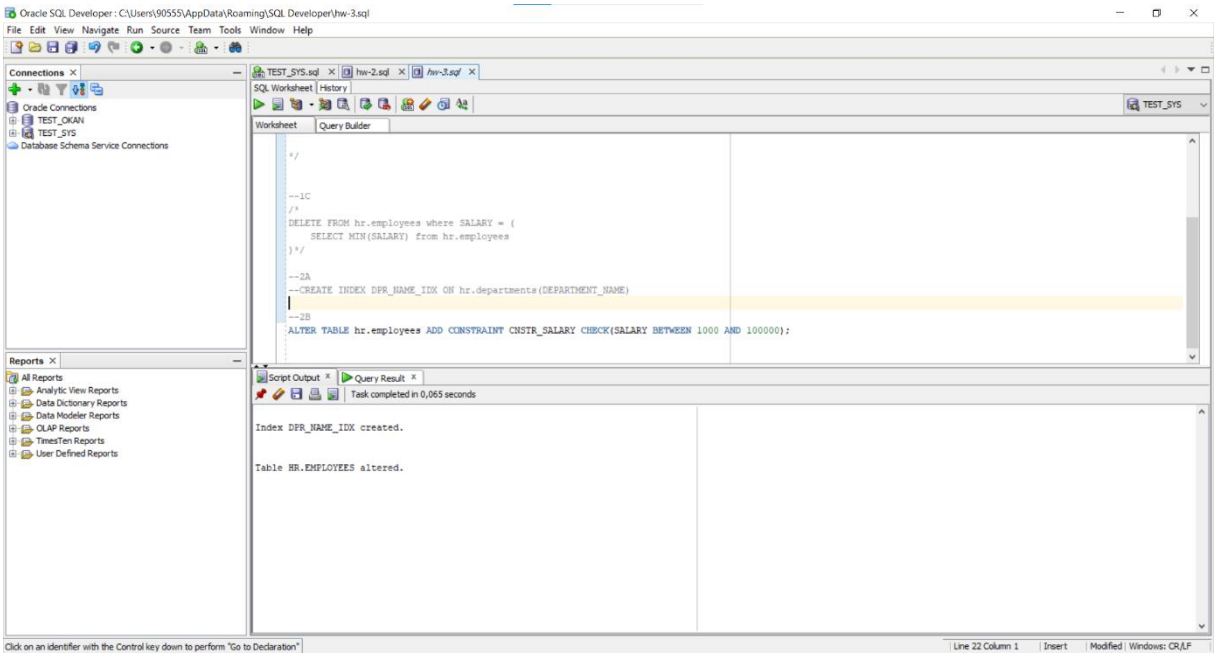
ALTER TABLE hr.employees ENABLE CONSTRAINT CNSTR_SALARY

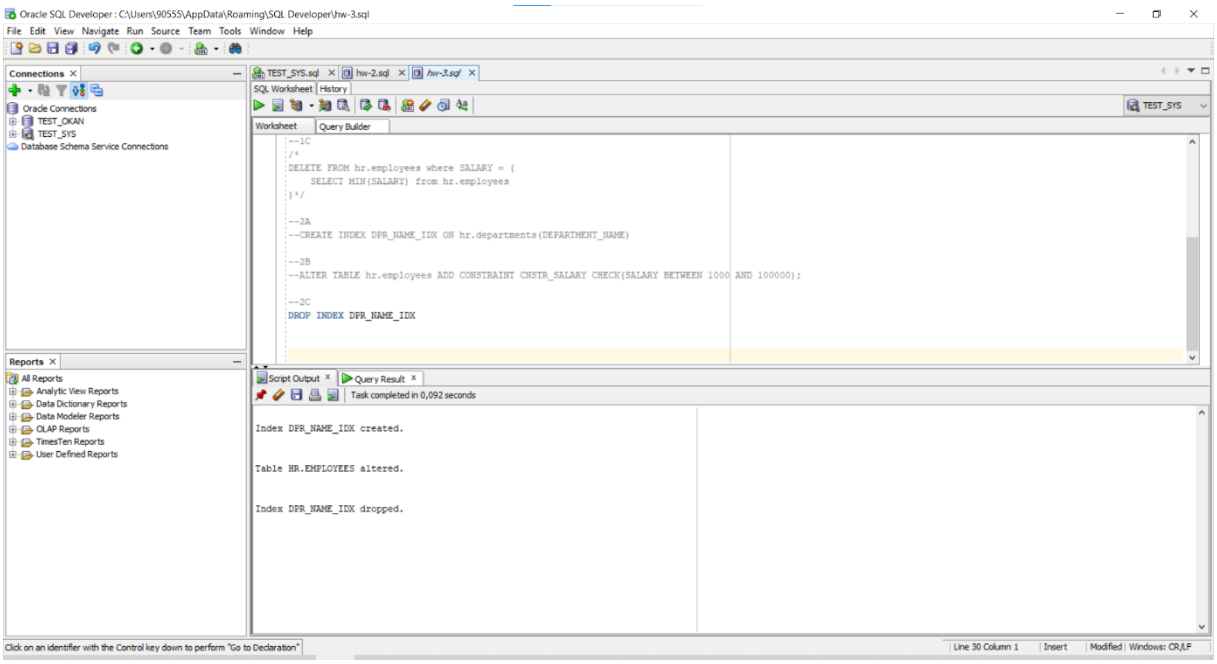
ALTER TABLE hr.employees DISABLE CONSTRAINT CNSTR_SALARY

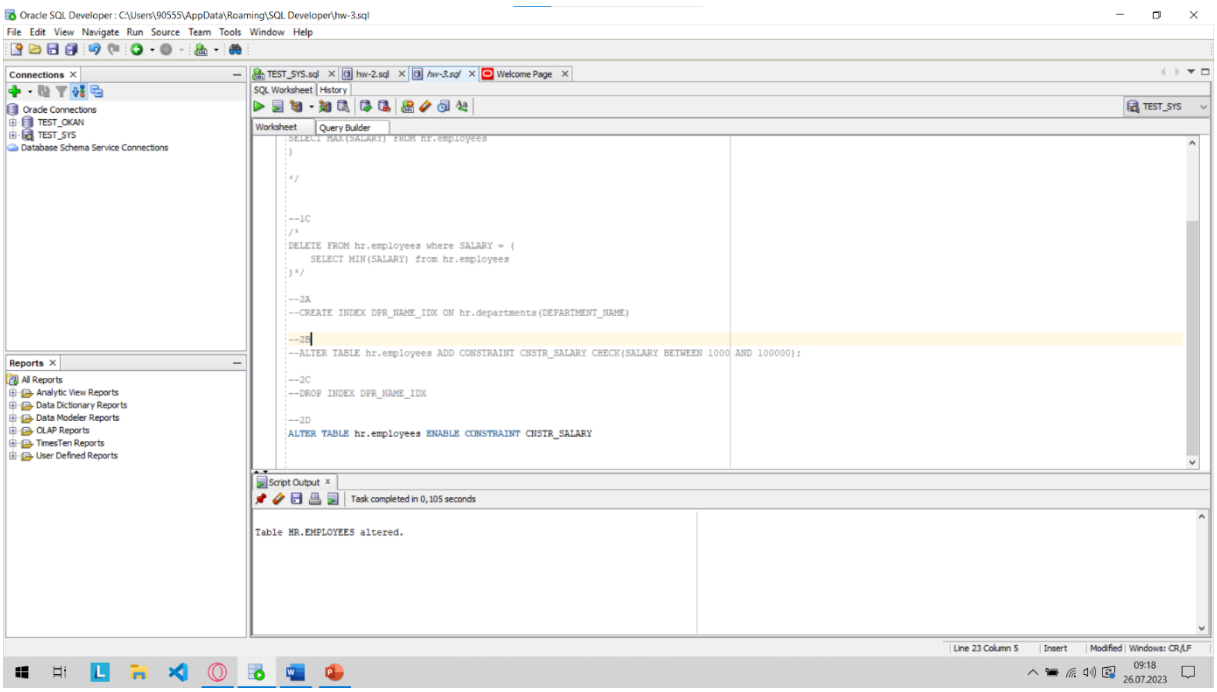
ALTER TABLE hr.employees DROP CONSTRAINT CNSTR_SALARY

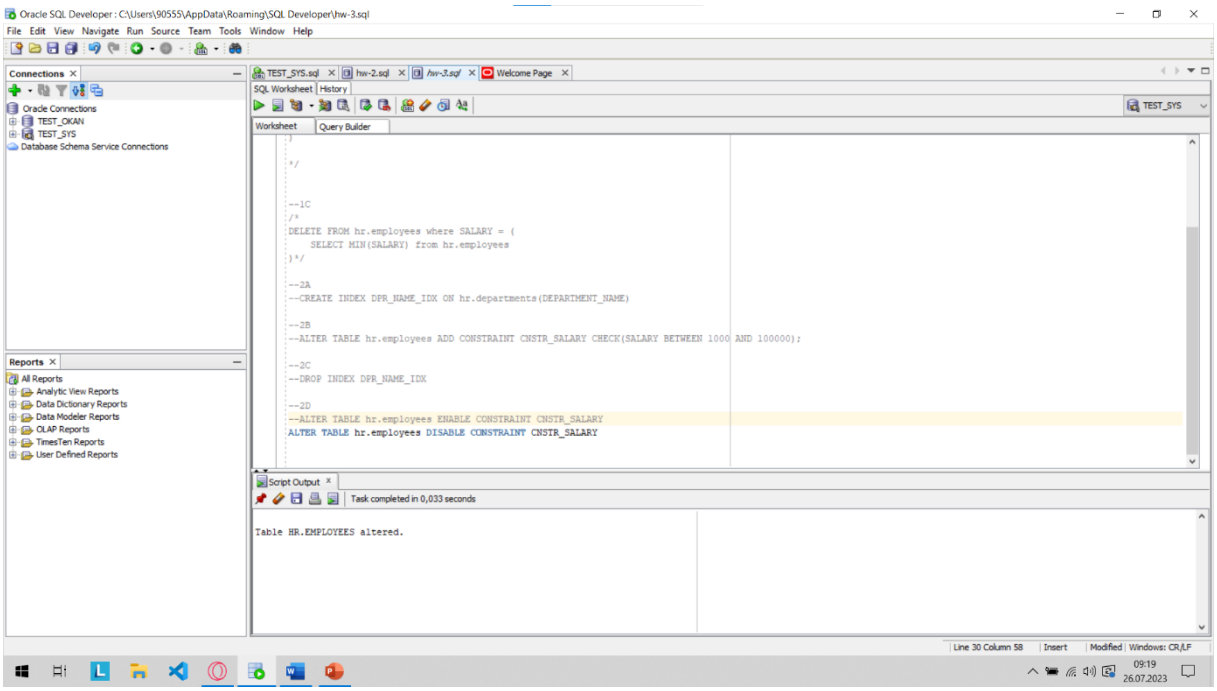
Screenshot:

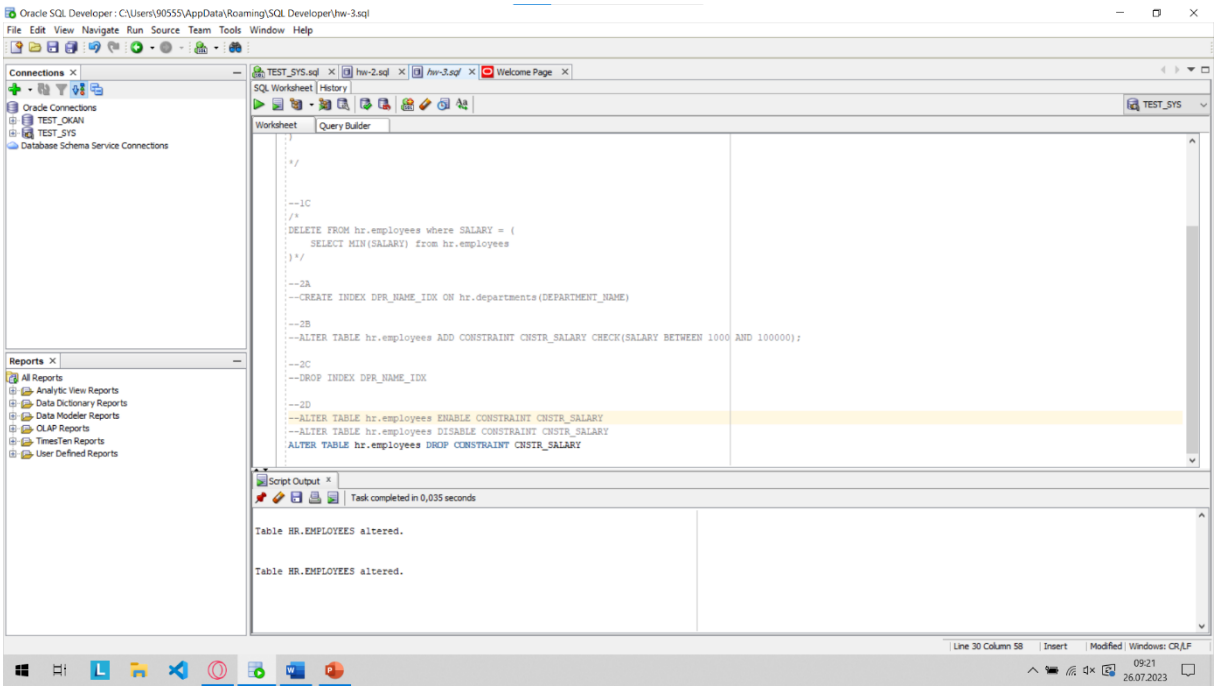





Oracle SQL Developer : C:\Users\90555\AppData\Roaming\SQL Developer\hw-3.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
TEST_OKAN
TEST_SYS
Database Schema Service Connections
TEST_SYS.sql
hw-2.sql
hw-3.sql
SQL Worksheet
History
Worksheet
Query Builder
--1C
/*
DELETE FROM hr.employees where SALARY = (
SELECT MIN(SALARY) from hr.employees
)*/
--2A
--CREATE INDEX DPR_NAME_IDX ON hr.departments(DEPARTMENT_NAME)
--2B
--ALTER TABLE hr.employees ADD CONSTRAINT CNSTR_SALARY CHECK(SALARY BETWEEN 1000 AND 100000);
--2C
DROP INDEX DPR_NAME_IDX
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Script Output
Query Result
Task completed in 0,092 seconds
Index DPR_NAME_IDX created.
Table HR.EMPLOYEES altered.
Index DPR_NAME_IDX dropped.
Click on an identifier with the Control key down to perform "Go to Declaration"
Line 30 Column 1
Insert
Modified
Windows: CR/LF


Oracle SQL Developer : C:\Users\90555\AppData\Roaming\SQL Developer\hw-3.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
TEST_OKAN
TEST_SYS
Database Schema Service Connections
TEST_SYS.sql
hw-2.sql
hw-3.sql
Welcome Page
SQL Worksheet
History
Worksheet
Query Builder
--1C
/*
DELETE FROM hr.employees where SALARY = (
SELECT MIN(SALARY) from hr.employees
)*/
--2A
--CREATE INDEX DPR_NAME_IDX ON hr.departments(DEPARTMENT_NAME)
--2B
--ALTER TABLE hr.employees ADD CONSTRAINT CNSTR_SALARY CHECK(SALARY BETWEEN 1000 AND 100000);
--2C
--DROP INDEX DPR_NAME_IDX
--2D
ALTER TABLE hr.employees ENABLE CONSTRAINT CNSTR_SALARY
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
Script Output
Task completed in 0,105 seconds
Table HR.EMPLOYEES altered.
Line 23 Column 5
Insert
Modified
Windows: CR/LF
09:18
26.07.2023


Oracle SQL Developer : C:\Users\90555\AppData\Roaming\SQL Developer\hw-3.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
TEST_OKAN
TEST_SYS
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
TEST_SYS.sql
hw-2.sql
hw-3.sql
Welcome Page
SQL Worksheet
History
Worksheet
Query Builder
*/
--1C
/*
DELETE FROM hr.employees where SALARY = (
SELECT MIN(SALARY) from hr.employees
)*/
--2A
--CREATE INDEX DPR_NAME_IDX ON hr.departments(DEPARTMENT_NAME)
--2B
--ALTER TABLE hr.employees ADD CONSTRAINT CNSTR_SALARY CHECK(SALARY BETWEEN 1000 AND 100000);
--2C
--DROP INDEX DPR_NAME_IDX
--2D
--ALTER TABLE hr.employees ENABLE CONSTRAINT CNSTR_SALARY
ALTER TABLE hr.employees DISABLE CONSTRAINT CNSTR_SALARY
Script Output
Task completed in 0,033 seconds
Table HR.EMPLOYEES altered.
Line 30 Column 58
Insert
Modified
Windows: CR/LF
09:19
26.07.2023


Oracle SQL Developer : C:\Users\90555\AppData\Roaming\SQL Developer\hw-3.sql
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
TEST_OKAN
TEST_SYS
Database Schema Service Connections
Reports
All Reports
Analytic View Reports
Data Dictionary Reports
Data Modeler Reports
OLAP Reports
TimesTen Reports
User Defined Reports
TEST_SYS.sql
hw-2.sql
hw-3.sql
Welcome Page
SQL Worksheet
History
Worksheet
Query Builder
*/
--1C
/*
DELETE FROM hr.employees where SALARY = (
SELECT MIN(SALARY) from hr.employees
)*/
--2A
--CREATE INDEX DPR_NAME_IDX ON hr.departments(DEPARTMENT_NAME)
--2B
--ALTER TABLE hr.employees ADD CONSTRAINT CNSTR_SALARY CHECK(SALARY BETWEEN 1000 AND 100000);
--2C
--DROP INDEX DPR_NAME_IDX
--2D
--ALTER TABLE hr.employees ENABLE CONSTRAINT CNSTR_SALARY
--ALTER TABLE hr.employees DISABLE CONSTRAINT CNSTR_SALARY
ALTER TABLE hr.employees DROP CONSTRAINT CNSTR_SALARY
Script Output
Task completed in 0,035 seconds
Table HR.EMPLOYEES altered.
Table HR.EMPLOYEES altered.
Line 30 Column 58
Insert
Modified
Windows: CR/LF
09:21
26.07.2023

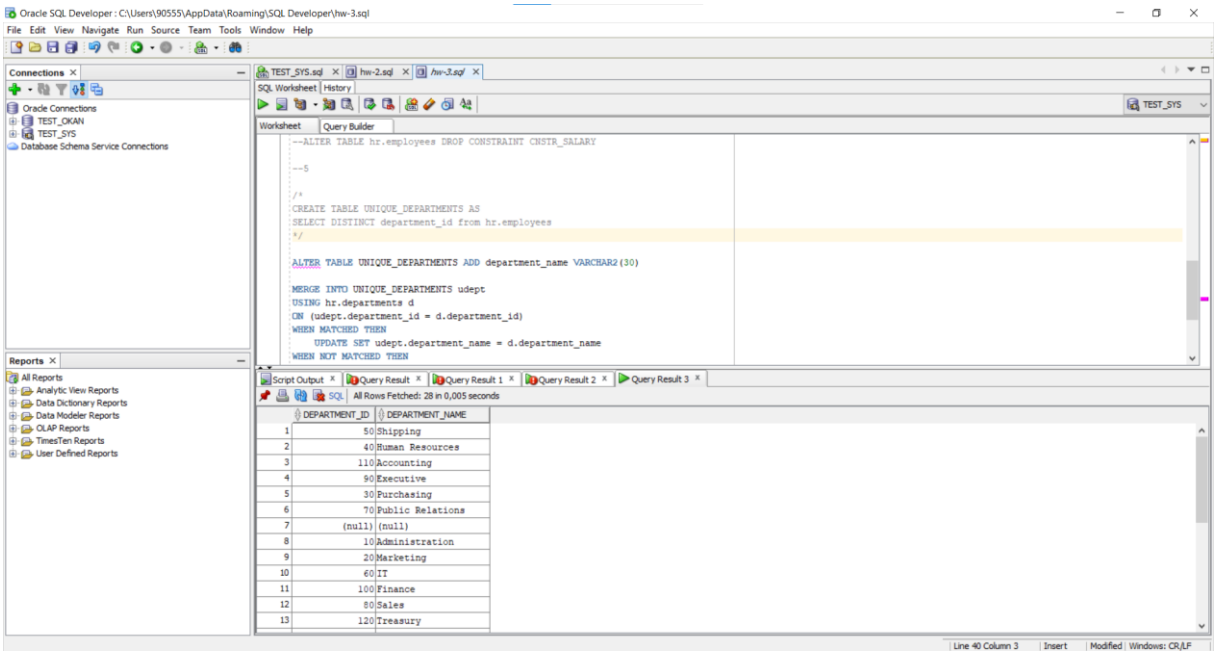
Exercise SQL03-EX-03:

Definiton : Create a table from EMPLOYEES with distinct department_id column. Add department_name to that table. With DEPARTMENTS table, update department_name for included department_ids and insert department_id and department_name values for not included rows. Use MERGE keyword.

SQL:

```
--create table

CREATE TABLE UNIQUE_DEPARTMENTS AS

SELECT DISTINCT department_id from hr.employees

--add column

ALTER TABLE UNIQUE_DEPARTMENTS ADD department_name VARCHAR2(30)

--update with merge keyword

MERGE INTO UNIQUE_DEPARTMENTS udept

USING hr.departments d

ON (udept.department_id = d.department_id)

WHEN MATCHED THEN

   UPDATE SET udept.department_name = d.department_name

WHEN NOT MATCHED THEN

   INSERT (department_id, department_name)

   VALUES(d.department_id, d.department_name);

--display record

SELECT * FROM unique_departments
```

Screenshot:



Exercise SQL03-EX-04:

Definiton : Using **WITH** keyword, do following jobs:

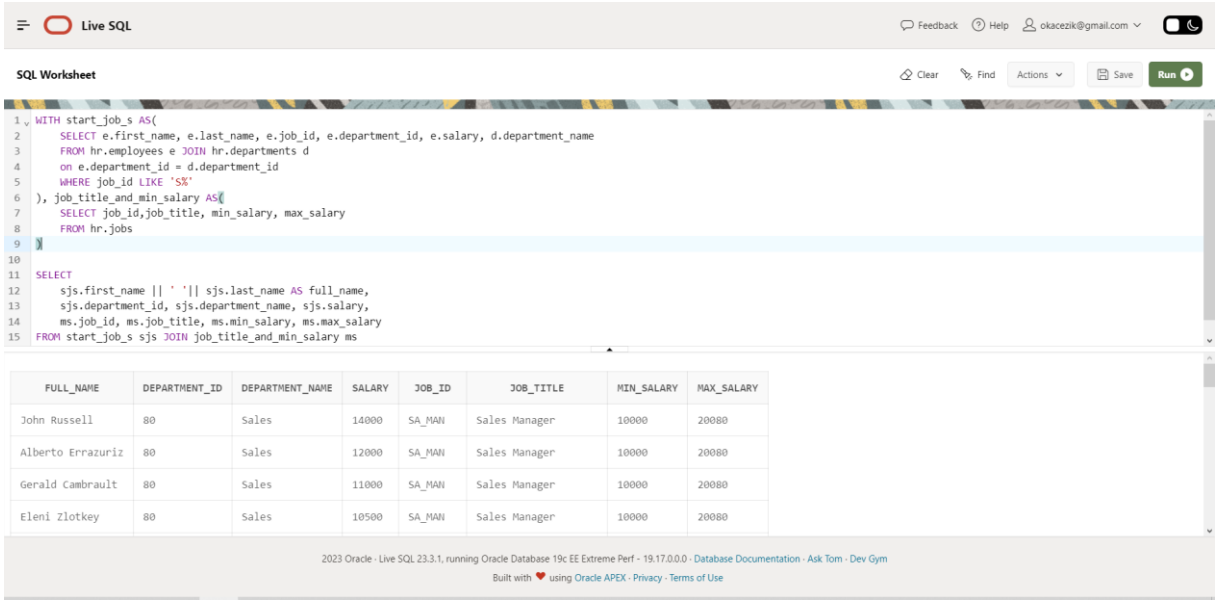
- Firstly select first_name, last_name, job_id, department_id from employees table whoes job_id starts with 'S'.
- Additionally select job_title and min-max salary amount.
- Add department_name to that query.
- Lastly concat first_name and last_name with space as full_name alias and list with other selected columns.

SQL:

```
WITH start_job_s AS(

   SELECT e.first_name, e.last_name, e.job_id, e.department_id, e.salary, d.department_name

   FROM hr.employees e JOIN hr.departments d

   on e.department_id = d.department_id

   WHERE job_id LIKE 'S%'

), job_title_and_min_salary AS(

   SELECT job_id,job_title, min_salary, max_salary
```

```
    FROM hr.jobs
)

SELECT
    sjs.first_name || ' '|| sjs.last_name AS full_name,
    sjs.department_id, sjs.department_name, sjs.salary,
    ms.job_id, ms.job_title, ms.min_salary, ms.max_salary
FROM start_job_s sjs JOIN job_title_and_min_salary ms
ON sjs.job_id = ms.job_id
```

Screenshot:



Exercise SQL03-EX-05:

Definiton : Search for COMMIT and ROLLBACK keywords and explain them.

SQL:

The **COMMIT** statement is used to permanently save the changes made during a transaction to the database.

```
BEGIN

 -- Transaction starts

 INSERT INTO UNIQUE_DEPARTMENTS (department_id, department_name)

 VALUES (6,'Doe');

 -- Commit the transaction to make changes permanent

 COMMIT;

END;
```

The **ROLLBACK** statement is used to undo the changes made during a transaction and restore the database to its state before the transaction started.

```
BEGIN

 -- Savepoint is created to mark the start of the transaction

 SAVEPOINT start_transaction;

 UPDATE UNIQUE_DEPARTMENTS SET department_id = 7 WHERE department_id = 6;

 -- Rollback to the savepoint to undo the changes

 ROLLBACK TO start_transaction;

END;
```

Screenshot:

```
BEGIN
  -- Transaction starts
  INSERT INTO UNIQUE_DEPARTMENTS (department_id, department_name)
  VALUES (6,'Doe');

  -- Commit the transaction to make changes permanent
  COMMIT;
END;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

Task completed in 0,158 seconds

```
Unknown Command:   COMMIT

Error starting at line : 60 in command -
 INSERT INTO UNIQUE_DEPARTMENTS(department_id, department_name) VALUES (3,'deneme3')
 INSERT INTO UNIQUE_DEPARTMENTS(department_id, department_name) VALUES (4,'deneme4')

 COMMIT
Error at Command Line : 61 Column : 9
Error report -
SQL Error: ORA-00933: SQL komutu tam doğru olarak sona ermedi
00933. 00000 -  "SQL command not properly ended"
*Cause:
*Action:

PL/SQL procedure successfully completed.
```

```
BEGIN
  -- Savepoint is created to mark the start of the transaction
  SAVEPOINT start_transaction;
  UPDATE UNIQUE_DEPARTMENTS SET department_id = 7 WHERE department_id = 6;
  -- Rollback to the savepoint to undo the changes
  ROLLBACK TO start_transaction;
END;
```

Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

Task completed in 0,039 seconds

```
 INSERT INTO UNIQUE_DEPARTMENTS(department_id, department_name) VALUES (3,'deneme3')
 INSERT INTO UNIQUE_DEPARTMENTS(department_id, department_name) VALUES (4,'deneme4')

 COMMIT
Error at Command Line : 61 Column : 9
Error report -
SQL Error: ORA-00933: SQL komutu tam doğru olarak sona ermedi
00933. 00000 -  "SQL command not properly ended"
*Cause:
*Action:

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```