

<b>Topic</b>	Oracle SQL Language Fundamentals I
<b>Document Name</b>	SQL01-EX-01-05



JULY – 2023

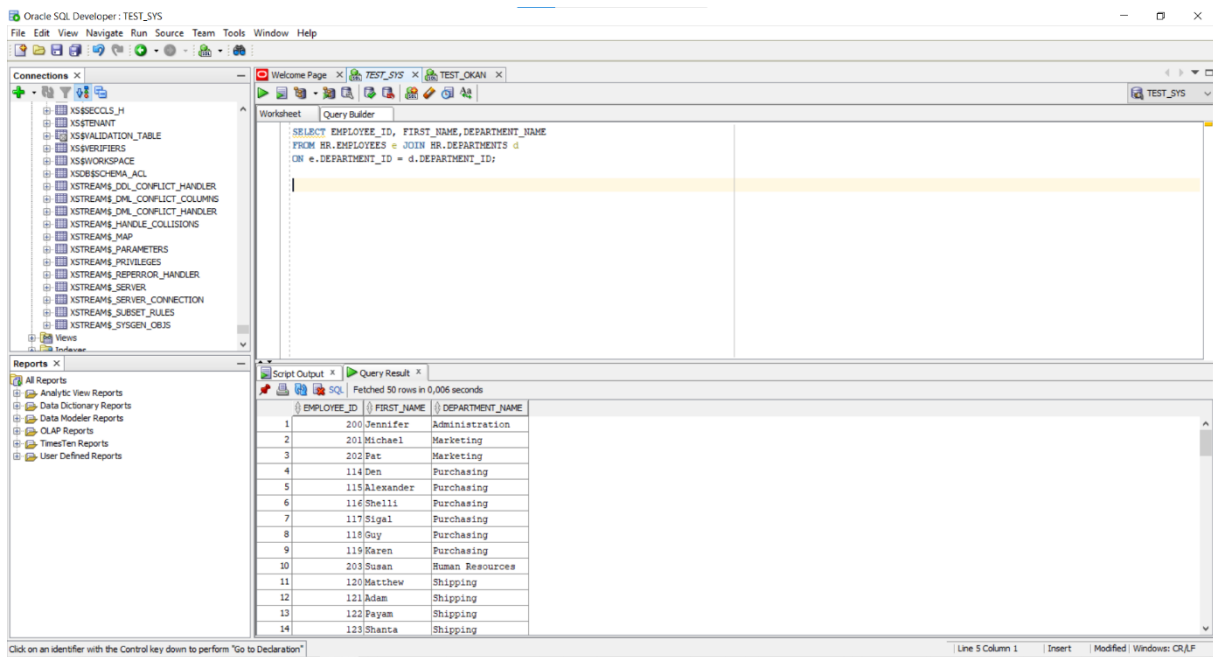
OKAN ÇEZİK

# Oracle SQL Language Fundamentals I

## Exercise SQL01-EX-01:

**Definiton :** Write an SQL query that selects employee's id, employee's first name and employee's department name for all employees. (Please use HR.EMPLOYEES and HR.DEPARTMENTS tables.)

### Output :



The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left lists various database connections. The 'Worksheet' pane in the center contains the following SQL query:

```
SELECT EMPLOYEE_ID, FIRST_NAME, DEPARTMENT_NAME
FROM HR.EMPLOYEES e JOIN HR.DEPARTMENTS d
ON e.DEPARTMENT_ID = d.DEPARTMENT_ID;
```

The 'Query Result' pane at the bottom displays the output of the query, showing 14 rows of data. The columns are EMPLOYEE\_ID, FIRST\_NAME, and DEPARTMENT\_NAME.

EMPLOYEE_ID	FIRST_NAME	DEPARTMENT_NAME
200	Jennifer	Administration
201	Michael	Marketing
202	Pat	Marketing
114	Den	Purchasing
115	Alexander	Purchasing
116	Stell	Purchasing
117	Sigal	Purchasing
118	Jay	Purchasing
119	Saren	Purchasing
203	Susan	Human Resources
120	Matthew	Shipping
121	Adam	Shipping
122	Payan	Shipping
123	Shanta	Shipping

## Exercise SQL01-EX-02:

**Definiton :** Create a report that displays the employee's id and their manager's id. (Please use HR.EMPLOYEES table)

## Output :

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for 'TEST\_SYS', including tables, views, and other objects. The main workspace shows a SQL query that filters employees where the manager ID is not null. The 'Query Result' pane displays the following data:

EMPLOYEE_ID	MANAGER_ID
101	100
102	100
103	102
104	103
105	103
106	103
107	103
108	101
109	108
110	108
111	108
112	108
113	108
114	100

## Exercise SQL01-EX-03:

**Definiton :** For example; first three character of PHONE\_NUMBER column gives us a operator of employee. Create a report that displays the operators and their total subscriber. But we want two different displays with diffrent queries. (Please use HR.EMPLOYEES table)

## Output :

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for 'TEST\_SYS'. The main workspace shows a SQL query that extracts the first three characters of the phone number as an operator and counts the number of employees for each operator. The 'Query Result' pane displays the following data:

OPERATOR	TOTAL
515	21
650	45
603	1
590	5
011	35

## Exercise SQL01-EX-04:

**Definiton :** Create a table (table name like HR.EMP) from HR.EMPLOYEES table. Insert a new row to HR.EMP table and update this employee's phone number and salary. Delete your new row and display the HR.EMP table. Finally drop your table HR.EMP.

### Output :

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for 'TEST\_SYS'. The main window shows the SQL Worksheet with the following SQL code:

```
/*  
WHERE MANAGER_ID IS NOT NULL  
*/  
  
/*  
SELECT SUBSTR(PHONE_NUMBER,1,3) AS Operator, COUNT(*) AS Total  
FROM HR.EMPLOYEES  
Group By SUBSTR(PHONE_NUMBER,1,3)  
*/  
  
CREATE TABLE HR.EMP  
(  
    EMPLOYEE_ID NUMBER PRIMARY KEY NOT NULL,  
    FIRST_NAME VARCHAR2(20) NOT NULL,  
    LAST_NAME VARCHAR2(30) NOT NULL,  
    EMAIL VARCHAR2(40) NOT NULL,  
    PHONE_NUMBER CHAR(11) NOT NULL,  
    HIRE_DATE DATE  
)
```

The Script Output pane shows the execution results:

```
Task completed in 0.123 seconds  
  
PHONE_NUMBER CHAR(11) NOT NULL,  
HIRE_DATE DATE  
)  
  
Error report -  
ORA-01031: Yetkililer Veterans  
01031. 00000 - "insufficient privileges"  
*Cause: An attempt was made to perform a database operation without  
the necessary privileges.  
*Action: Ask your database administrator or designated security  
administrator to grant you the necessary privileges  
  
Table HR.EMP created.
```

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for 'TEST\_SYS'. The main window shows the SQL Worksheet with the following SQL code:

```
/*  
SELECT SUBSTR(PHONE_NUMBER,1,3) AS Operator, COUNT(*) AS Total  
FROM HR.EMPLOYEES  
Group By SUBSTR(PHONE_NUMBER,1,3)  
*/  
  
CREATE TABLE HR.EMP  
(  
    EMPLOYEE_ID NUMBER PRIMARY KEY NOT NULL,  
    FIRST_NAME VARCHAR2(20) NOT NULL,  
    LAST_NAME VARCHAR2(30) NOT NULL,  
    EMAIL VARCHAR2(40) NOT NULL,  
    PHONE_NUMBER CHAR(11) NOT NULL,  
    HIRE_DATE DATE  
)  
  
INSERT INTO HR.EMP (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE)  
VALUES(1, 'Okan', 'Cezik', 'okacezik@gmail.com', '05550651690', SYSDATE)
```

The Script Output pane shows the execution results:

```
Task completed in 0.046 seconds  
  
VALUES(1, 'Okan', 'Cezik', 'okacezik@gmail.com', '05550651690', SYSDATE)  
  
Error at Command Line : 32 Column : 40  
Error report -  
SQL Error: ORA-01861: sabit deęer format dizeyle eęleęmiyor  
01861. 00000 - "literal does not match format string"  
*Cause: Literals in the input must be the same length as literals in  
the format string (with the exception of leading whitespace). If the  
"FX" modifier has been toggled on, the literal must match exactly,  
with no extra whitespace.  
*Action: Correct the format string to match the literal.  
  
1 row inserted.
```

Oracle SQL Developer: C:\Users\90555\Desktop\sqlhw-1\TEST\_SYS.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections X

- TEST\_OKAN
- TEST\_SYS
  - 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

Reports X

- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimeTen Reports
- User Defined Reports

TEST\_SYS.sql

Worksheet Query Builder

```

EMPLOYEE_ID NUMBER PRIMARY KEY NOT NULL,
FIRST_NAME VARCHAR2(20) NOT NULL,
LAST_NAME VARCHAR2(30) NOT NULL,
EMAIL VARCHAR2(40) NOT NULL,
PHONE_NUMBER CHAR(11) NOT NULL,
HIRE_DATE DATE
)
*/

INSERT INTO HR.EMP (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE)
VALUES(1, 'Okan', 'Cesik', 'okacezik@gmail.com', '05550651690', SYSDATE);
*/

UPDATE HR.EMP
SET PHONE_NUMBER = '05555555555'
WHERE EMPLOYEE_ID = 1;
  
```

Script Output Query Result

Task completed in 0.063 seconds

which is too wide for the width of the destination column. The name of the column is given, along with the actual width of the value, and the maximum allowed width of the column. Note that widths are reported in characters if / semantics are in effect for the column, otherwise widths are reported in bytes.

Action: Examine the SQL statement for correctness. Check source and destination column data types. Either make the destination column wider, or use a subset of the source column (i.e. use substring).

1 row updated.

Saved: C:\Users\90555\Desktop\sqlhw-1\TEST\_SYS.sql

Line 38 Column 23 Insert Modified Windows: CR,LF

16:28 22.07.2023

Oracle SQL Developer: C:\Users\90555\Desktop\sqlhw-1\TEST\_SYS.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections X

- TEST\_OKAN
- TEST\_SYS
  - 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

Reports X

- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimeTen Reports
- User Defined Reports

TEST\_SYS.sql

Worksheet Query Builder

```

CREATE TABLE HR.EMP
(
  EMPLOYEE_ID NUMBER PRIMARY KEY NOT NULL,
  FIRST_NAME VARCHAR2(20) NOT NULL,
  LAST_NAME VARCHAR2(30) NOT NULL,
  EMAIL VARCHAR2(40) NOT NULL,
  PHONE_NUMBER CHAR(11) NOT NULL,
  HIRE_DATE DATE
)
*/

INSERT INTO HR.EMP (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE)
VALUES(1, 'Okan', 'Cesik', 'okacezik@gmail.com', '05550651690', SYSDATE);
*/

DELETE FROM HR.EMP
WHERE EMPLOYEE_ID = 1;
  
```

Script Output Query Result

Task completed in 0.045 seconds

Note that widths are reported in characters if character length semantics are in effect for the column, otherwise widths are reported in bytes.

Action: Examine the SQL statement for correctness. Check source and destination column data types. Either make the destination column wider, or use a subset of the source column (i.e. use substring).

1 row updated.

1 row deleted.

Saved: C:\Users\90555\Desktop\sqlhw-1\TEST\_SYS.sql

Line 37 Column 12 Insert Windows: CR,LF

16:29 22.07.2023

## Exercise SQL01-EX-05:

**Definiton :** Select employees' first name and last name as masked with "\*" character as shown in sample output below.

### Output :

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL script in the 'Worksheet' tab. The script includes a comment, a DELETE statement, and a SELECT statement that masks the first and last names of employees using the SUBSTR function and asterisks. The 'Script Output' tab shows the results of the query, displaying 11 rows of masked names.

```
SET PHONE_NUMBER = '05555555555'
WHERE EMPLOYEE_ID = 1;

/*
DELETE FROM HR.EMP
WHERE EMPLOYEE_ID = 1
*/

--DROP TABLE HR.EMP

SELECT
  READ(SUBSTR(FIRST_NAME, 1, 1), LENGTH(FIRST_NAME), '*')
  ||
  || READ(SUBSTR(LAST_NAME, 1, 1), LENGTH(LAST_NAME), '*')
  AS CUSTOMER_NAME
FROM HR.EMPLOYEES;
```

	CUSTOMER_NAME
1	E***** A****
2	D***** A****
3	M***** A*****
4	D***** A*****
5	S***** B****
6	S***** B****
7	A**** B*****
8	E***** B*****
9	S***** B****
10	D***** B*****
11	T***** B*****

Line 57 Column 1 | Insert | Modified | Windows: CR/LF