|  |  |
| --- | --- |
| **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:**

ALTER TABLE EMPLOYEES ADD MAX\_SALARY NUMBER(8,2);

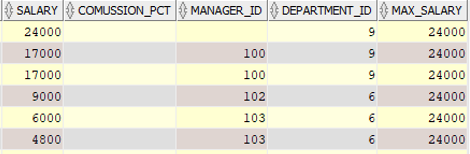
UPDATE EMPLOYEES SET MAX\_SALARY = (SELECT MAX(SALARY) FROM EMPLOYEES);

DELETE FROM EMPLOYEES WHERE SALARY =(SELECT MIN(SALARY) FROM EMPLOYEES);

SELECT \* FROM EMPLOYEES;

**Screenshot:**







## 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:**

**CREATE INDEX DPR\_NAME\_IDX ON DEPARTMENTS(DEPARTMENT\_NAME);**

**ALTER TABLE EMPLOYEES ADD CONSTRAINT CNSTR\_SALARY CHECK (SALARY>1000 AND SALARY<100000) ;**

**DROP INDEX DPR\_NAME\_IDX;**

**ALTER TABLE EMPLOYEES ENABLE CONSTRAINT CNSTR\_SALARY;**

**ALTER TABLE EMPLOYEES DISABLE CONSTRAINT CNSRTR\_SALARY;**

**ALTER TABLE EMPLOYEES DROP CONSTRAINT CNSRTR\_SALARY;**

**Screenshot:**









## 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 NEW\_EMP AS SELECT DISTINCT DEPARTMENT\_ID FROM EMPLOYEES;

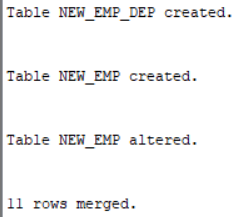
ALTER TABLE NEW\_EMP ADD DEPARTMENT\_NAME VARCHAR2(20);

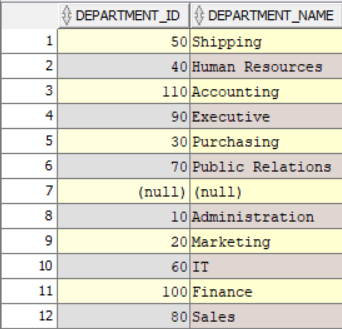
MERGE INTO NEW\_EMP USING DEPARTMENTS ON (NEW\_EMP.DEPARTMENT\_ID = DEPARTMENTS.DEPARTMENT\_ID) WHEN MATCHED THEN UPDATE SET NEW\_EMP.DEPARTMENT\_NAME=DEPARTMENTS.DEPARTMENT\_NAME;

SELECT \* FROM NEW\_EMP;\_NAME;

SELECT \* FROM NEW\_EMP;

**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 EMP\_JOBS AS (SELECT FIRST\_NAME,LAST\_NAME,JOB\_ID,DEPARTMENT\_ID FROM EMPLOYEES WHERE FIRST\_NAME LIKE 'S%'),

JOB\_MAX\_MIN AS ( SELECT JOB\_ID,JOB\_TITLE,MIN\_SALARY,MAX\_SALARY FROM JOBS),

DEP\_DET AS ( SELECT DEPARTMENT\_ID,DEPARTMENT\_NAME FROM DEPARTMENTS)

SELECT E.FIRST\_NAME,E.LAST\_NAME,E.JOB\_ID,

JM.JOB\_TITLE,JM.MAX\_SALARY,JM.MIN\_SALARY,

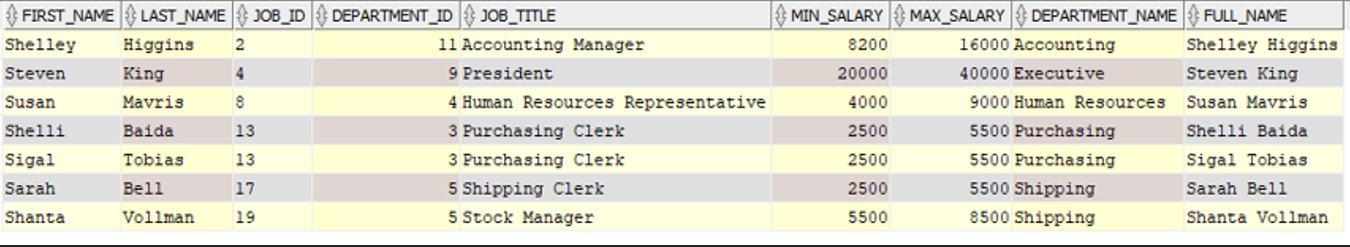
D.DEPARTMENT\_NAME,

CONCAT(E.FIRST\_NAME,E.LAST\_NAME) AS FULL\_NAME FROM EMP\_JOBS E

JOIN JOB\_MAX\_MIN JM ON E.JOB\_ID = JM.JOB\_ID

JOIN DEP\_DET D ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID;

**Screenshot:**



## Exercise SQL03-EX-05:

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

**SQL:**

**Commit is the command used to persist transactions in the database.**

**Rollback lets you undo your actions in a database**

**Screenshot:**