|  |  |
| --- | --- |
| **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 HR.employees ADD (Max\_Salary NUMBER (8));

UPDATE HR.EMPLOYEES

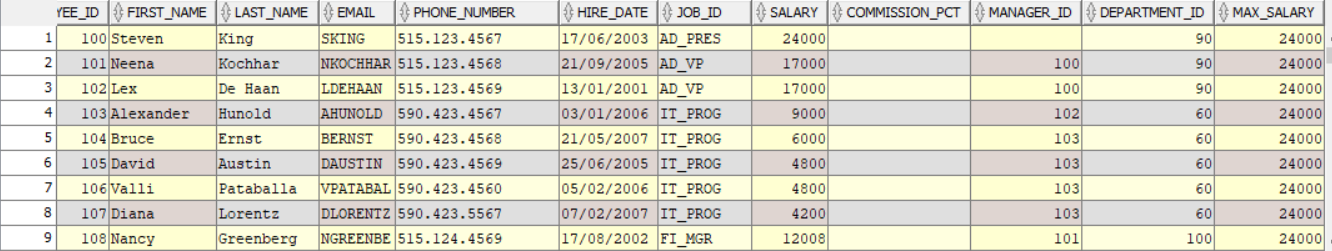
SET Max\_Salary = (SELECT MAX(salary) FROM HR.employees

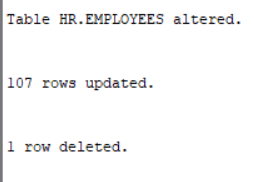
);

DELETE FROM HR.employees

WHERE salary = (SELECT MIN(salary) FROM HR.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 HR.departments (department\_name);

ALTER TABLE HR.employees ADD CONSTRAINT CNSTR\_SALARY

CHECK (salary>1000 AND salary<100000);

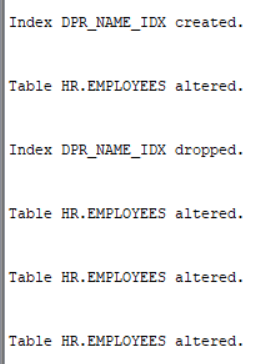
DROP INDEX DPR\_NAME\_IDX;

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:**



## 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\_TABLE AS (SELECT department\_id FROM HR.departments);

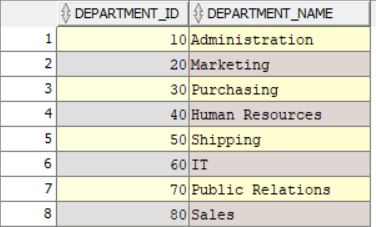
ALTER TABLE NEW\_TABLE ADD (department\_name VARCHAR2(30));

MERGE INTO NEW\_TABLE t USING (SELECT \* FROM HR.departments) d ON

(t.department\_id = d.department\_id) WHEN MATCHED THEN

UPDATE SET t.department\_name = d.department\_name;

**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 MY\_EMPLOYEES AS (SELECT FIRST\_NAME,LAST\_NAME,JOB\_ID,DEPARTMENT\_ID

FROM HR.EMPLOYEES WHERE FIRST\_NAME LIKE 'S%'),

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

DEP\_DETAIL AS (SELECT DEPARTMENT\_ID,DEPARTMENT\_NAME FROM HR.DEPARTMENTS)

SELECT

e.first\_name,

e.last\_name,

e.job\_id,

e.department\_id,

j.job\_title,

j.min\_salary,

j.max\_salary,

d.department\_name,

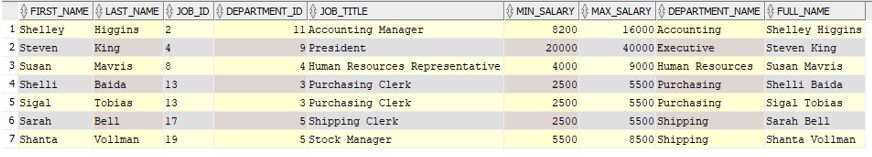
e.first\_name || ' ' || e.last\_name AS full\_name

FROM MY\_EMPLOYEES e

JOIN JOB\_MAX\_MIN j ON e.job\_id = j.job\_id

JOIN DEP\_DETAIL d ON e.department\_id = d.department\_id;

**Screenshot:**



## Exercise SQL03-EX-05:

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

The “COMMIT” statement is used to permanently save the changes made in a transaction to the database permanently.

The “ROLLBACK” statement is used to undo the changes made in a transaction and return the database to previous state before the transaction began.

**SQL:**

COMMIT

ROLLBACK

**Screenshot:**

