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

create table emp as select \* from hr.employees;

select \* from emp ;

alter table emp add (max\_salary number);

ALTER TABLE emp RENAME COLUMN max\_salary TO maximum\_salary;

update emp set maximum\_salary = (select max(Salary) from emp);

DELETE FROM emp WHERE salary = (SELECT MIN(salary) FROM emp);

**Screenshot:**

ekran görüntüsü, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

## 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 table depart as select \* from hr.departments;

select \* from depart;

CREATE INDEX DPR\_NAME\_IDX ON depart (DEPARTMENT\_NAME);

ALTER TABLE emp

ADD CONSTRAINT CNSTR\_SALARY

CHECK (SALARY BETWEEN 1000 AND 100000);

drop index DPR\_NAME\_IDX ;

ALTER TABLE emp

ENABLE CONSTRAINT CNSTR\_SALARY;

ALTER TABLE emp

DISABLE CONSTRAINT CNSTR\_SALARY;

ALTER TABLE emp

DROP CONSTRAINT CNSTR\_SALARY;

## 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:**

**MERGE INTO emp1 em**

**using hr.departments dp**

**on (em.department\_id = dp.department\_id )**

**when matched then**

**UPDATE SET em.department\_name = dp.department\_name**

**when not matched then**

**insert ( department\_id , department\_name )**

**values(dp.department\_id , 'null');**

**Screenshot:**

metin, ekran görüntüsü, multimedya yazılımı, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

## 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:**

**SELECT distinct**

**emp.first\_name || ' ' || emp.last\_name AS full\_name,**

**emp.department\_id,**

**jobs.job\_title,**

**MAX(emp.salary) AS max\_salary,**

**MIN(emp.salary) AS min\_salary,**

**dep.department\_name**

**FROM**

**hr.employees emp**

**JOIN**

**hr.jobs jobs ON emp.job\_id = jobs.job\_id**

**JOIN**

**hr.departments dep ON emp.department\_id = dep.department\_id**

**WHERE**

**jobs.job\_id LIKE 'S%'**

**GROUP BY**

**emp.first\_name,**

**emp.last\_name,**

**emp.department\_id,**

**jobs.job\_title,**

**dep.department\_name;**

ekran görüntüsü, metin, multimedya yazılımı, grafik yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu**Screenshot:**

## Exercise SQL03-EX-05:

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

**Commit** records all transactions made permanently to the database.

**Rollback** command undoes the changes made during the current transaction. It restores the database to its previous state, thus recovering from incorrectly performed transactions.