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

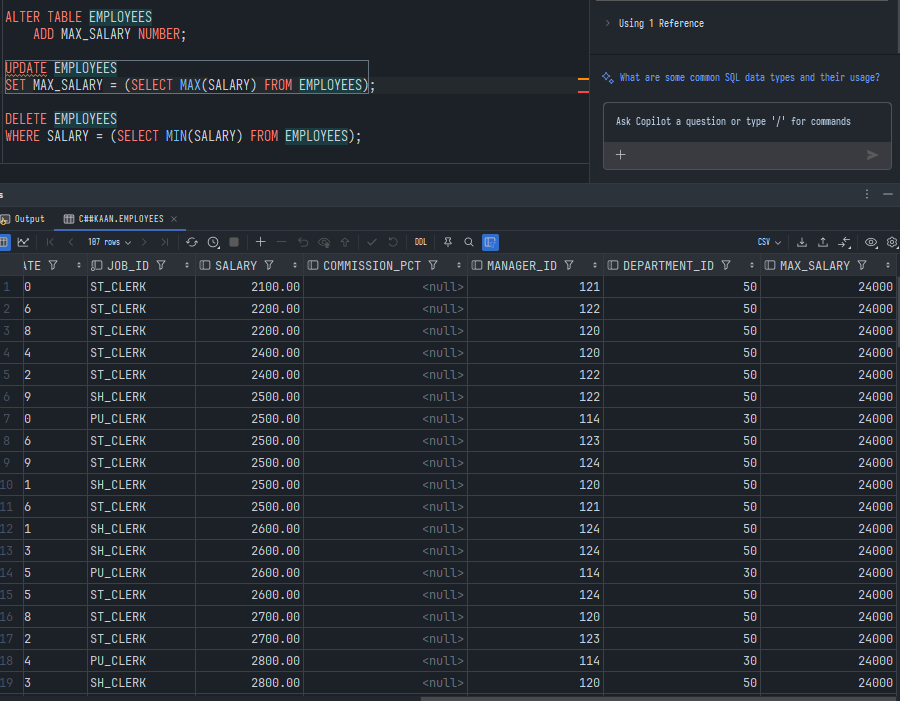
## Exercise SQL03-EX-01:

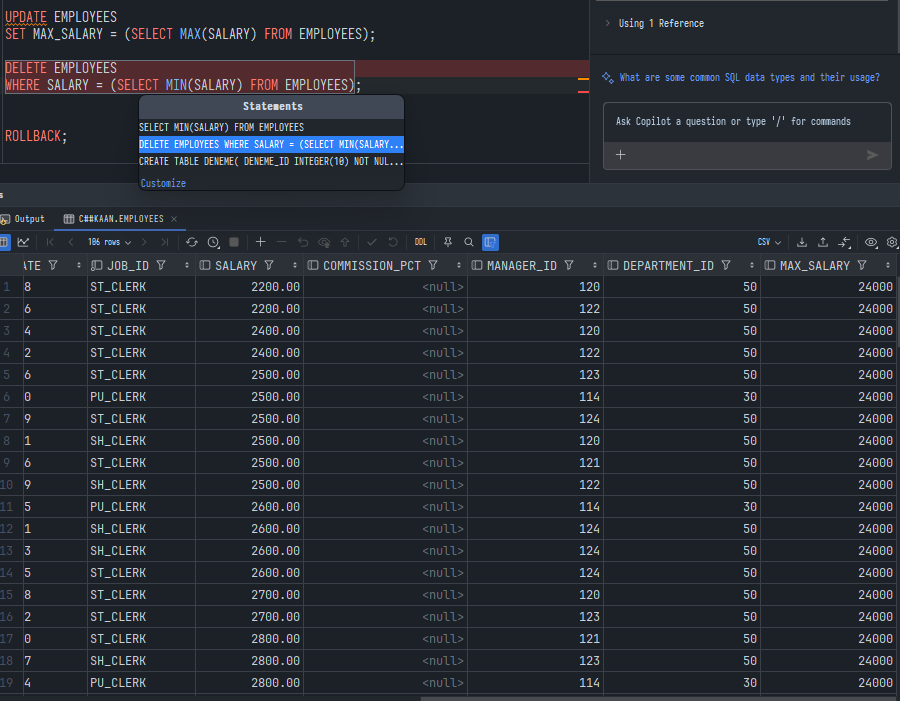
**Definiton :** Write followig SQL queries:

* 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;  
  
UPDATE EMPLOYEES  
SET MAX\_SALARY = (SELECT MAX(SALARY) FROM EMPLOYEES);  
  
DELETE EMPLOYEES  
WHERE SALARY = (SELECT MIN(SALARY) FROM EMPLOYEES);

**Screenshot:**



## Exercise SQL03-EX-02:

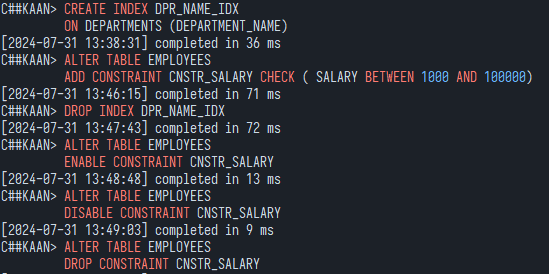
**Definiton :** Write followig SQL queries:

* 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:**

--1  
CREATE INDEX DPR\_NAME\_IDX  
ON DEPARTMENTS (DEPARTMENT\_NAME);  
  
-- 2  
  
ALTER TABLE EMPLOYEES  
ADD CONSTRAINT CNSTR\_SALARY CHECK ( SALARY BETWEEN 1000 AND 100000);  
  
--3  
DROP INDEX DPR\_NAME\_IDX;  
  
--4  
ALTER TABLE EMPLOYEES  
ENABLE CONSTRAINT CNSTR\_SALARY;  
  
ALTER TABLE EMPLOYEES  
DISABLE CONSTRAINT CNSTR\_SALARY;  
  
ALTER TABLE 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 EMPLOYEES\_DEPT AS  
SELECT DISTINCT DEPARTMENT\_ID  
FROM EMPLOYEES;  
  
ALTER TABLE EMPLOYEES\_DEPT  
ADD DEPARTMENT\_NAME VARCHAR2(256);  
  
MERGE INTO EMPLOYEES\_DEPT ED  
USING DEPARTMENTS D  
ON (ED.DEPARTMENT\_ID = D.DEPARTMENT\_ID)  
WHEN MATCHED THEN  
 UPDATE SET ED.DEPARTMENT\_NAME = D.DEPARTMENT\_NAME  
WHEN NOT MATCHED THEN  
 INSERT (DEPARTMENT\_ID, DEPARTMENT\_NAME)  
 VALUES (D.DEPARTMENT\_ID, D.DEPARTMENT\_NAME);

**Screenshot:**

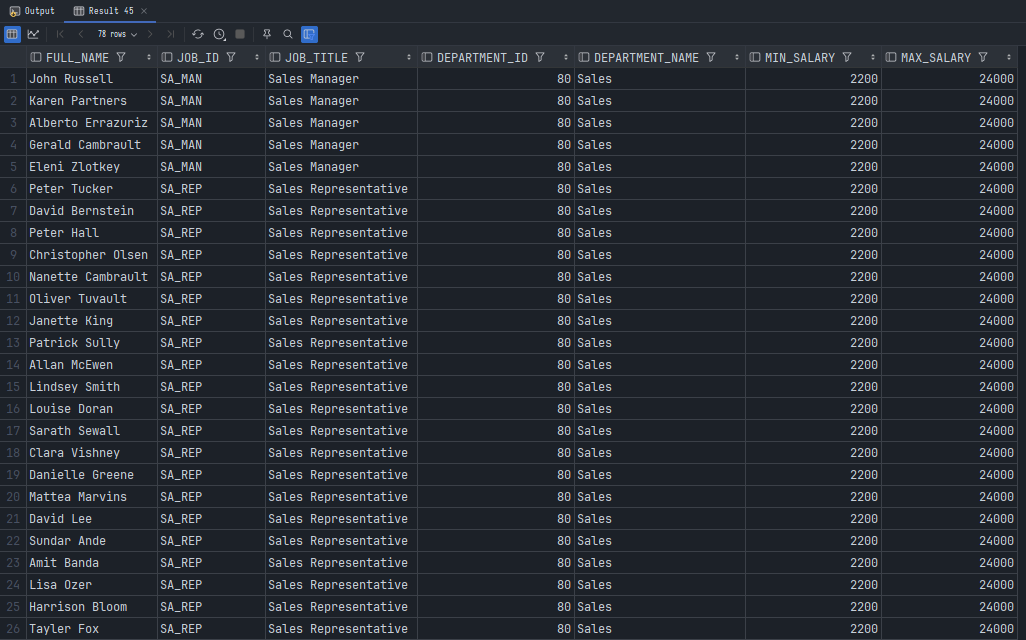
## Exercise SQL03-EX-04:

**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:**

SELECT  
 EMP.FIRST\_NAME || ' ' || EMP.LAST\_NAME AS FULL\_NAME,  
 EMP.JOB\_ID,  
 J.JOB\_TITLE,  
 EMP.DEPARTMENT\_ID,  
 D.DEPARTMENT\_NAME,  
 (SELECT MIN(SALARY) FROM EMPLOYEES) AS MIN\_SALARY,  
 (SELECT MAX(SALARY) FROM EMPLOYEES) AS MAX\_SALARY  
FROM  
 EMPLOYEES EMP  
 INNER JOIN JOBS J ON EMP.JOB\_ID = J.JOB\_ID  
 INNER JOIN DEPARTMENTS D ON EMP.DEPARTMENT\_ID = D.DEPARTMENT\_ID  
WHERE  
 EMP.JOB\_ID LIKE 'S%';

**Screenshot:**

## Exercise SQL03-EX-05:

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

*COMMIT:* The COMMIT statement in Oracle is used to save all the changes made in the current transaction to the database. Once a COMMIT is executed, the changes become permanent and are visible to other users. The syntax for committing a transaction is simply COMMIT;

*ROLLBACK:* The ROLLBACK statement in Oracle is used to undo all the changes made in the current transaction. It reverts the database to the state it was in before the transaction began, effectively canceling any data modifications such as INSERT, UPDATE, or DELETE operations performed during the transaction. This is useful for error recovery or if the transaction needs to be aborted for any reason. The syntax for rolling back a transaction is ROLLBACK;