Topic	Oracle SQL Language Fundamentals I
Document Name	SQL03-EX-01-05
	Enes Tahtacı

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.

Screenshots:

Worksheet | Query Builder

```sql
-- SQL03-EX-01:
ALTER TABLE employees
ADD max_salary NUMBER;

UPDATE employees
SET max_salary = (SELECT MAX(salary) FROM employees);

SELECT * FROM employees
WHERE salary = (SELECT MIN(salary) FROM employees);

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

SELECT * FROM employees;
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.004 seconds

	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMI
1	TJ	Olson	TJOLSON	650.124.8234	10-APR-07	ST_CLERK	2100	

Worksheet | Query Builder

```sql
-- SQL03-EX-01:
ALTER TABLE employees
ADD max_salary NUMBER;

UPDATE employees
SET max_salary = (SELECT MAX(salary) FROM employees);

SELECT * FROM employees
WHERE salary = (SELECT MIN(salary) FROM employees);

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

SELECT * FROM employees;
```

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.003 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	128	Steven	Markle	SMARKLE	650.124.1434	08-MAR-08	ST_CLERK	2200
2	136	Hazel	Philtanker	HPHILTAN	650.127.1634	06-FEB-08	ST_CLERK	2200

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.

Screenshot:

```
-- SQL03-EX-02:
CREATE INDEX dpr_name_idx
ON departments (department_name);

ALTER TABLE employees
ADD CONSTRAINT cnstr_salary
CHECK (salary BETWEEN 1000 AND 100000);

DROP INDEX dpr_name_idx;

ALTER TABLE employees
ENABLE CONSTRAINT cnstr_salary;

ALTER TABLE employees
DISABLE CONSTRAINT cnstr_salary;

ALTER TABLE employees
DROP CONSTRAINT cnstr_salary;
```

Script Output x Query Result x

Task completed in 0.023 seconds

```
Index DPR_NAME_IDX created.


Table EMPLOYEES altered.


Index DPR_NAME_IDX dropped.


Table EMPLOYEES altered.


Table EMPLOYEES altered.


Table EMPLOYEES altered.
```

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.

Screenshot:

Worksheet | Query Builder

```
-- SQL03-EX-03:
CREATE TABLE emp2 AS
SELECT DISTINCT department_id
FROM employees;

ALTER TABLE emp2 ADD(
    department_name  VARCHAR2(30)
);

UPDATE emp2
SET department_name = (SELECT department_name
                       FROM departments
                       WHERE emp2.department_id = departments.department_id)
WHERE department_id IN (SELECT department_id FROM emp2);

SELECT department_name,department_id FROM departments;

MERGE INTO emp2 e
USING departments d
ON (e.department_id = d.department_id)
WHEN MATCHED THEN
    UPDATE SET e.department_name = d.department_name
WHEN NOT MATCHED THEN
    INSERT (e.department_id, e.department_name)
    VALUES (d.department_id, d.department_name);
```

Script Output | Query Result

SQL | All Rows Fetched: 28 in 0.002 seconds

	DEPARTMENT_ID	DEPARTMENT_NAME
1	50	Shipping
2	40	Human Resources
3	110	Accounting
4	90	Executive
5	30	Purchasing
6	70	Public Relations
7	(null)	(null)
8	10	Administration
9	20	Marketing
10	60	IT
11	100	Finance
12	80	Sales
13	210	IT Support
14	200	Operations
15	270	Payroll

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.

Screenshot:

Worksheet | Query Builder

```sql
-- SQL03-EX-04:
WITH selected_employees AS (
    SELECT e.first_name, e.last_name, e.job_id, e.department_id
    FROM employees e
    WHERE e.job_id LIKE 'S%'
),
job_info AS (
    SELECT j.job_id, j.job_title, j.min_salary, j.max_salary
    FROM jobs j
),
employee_details AS (
    SELECT se.first_name, se.last_name, se.job_id, se.department_id,
           ji.job_title, ji.min_salary, ji.max_salary
    FROM selected_employees se
    JOIN job_info ji ON se.job_id = ji.job_id
)
SELECT ed.first_name || ' ' || ed.last_name AS full_name,
       ed.job_id, ed.department_id, ed.job_title,
       ed.min_salary, ed.max_salary, d.department_name
FROM employee_details ed
JOIN departments d ON ed.department_id = d.department_id;
```

Script Output | Query Result

SQL | Fetched 50 rows in 0.008 seconds

	FULL_NAME	JOB_ID	DEPARTMENT_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY	DEPARTMENT_NAME
1	Matthew Weiss	ST_MAN	50	Stock Manager	5500	8500	Shipping
2	Adam Fripp	ST_MAN	50	Stock Manager	5500	8500	Shipping
3	Payam Kaufling	ST_MAN	50	Stock Manager	5500	8500	Shipping
4	Shanta Vollman	ST_MAN	50	Stock Manager	5500	8500	Shipping
5	Kevin Mourgos	ST_MAN	50	Stock Manager	5500	8500	Shipping
6	Julia Nayer	ST_CLERK	50	Stock Clerk	2008	5000	Shipping
7	Irene Mikkilineni	ST_CLERK	50	Stock Clerk	2008	5000	Shipping
8	James Landry	ST_CLERK	50	Stock Clerk	2008	5000	Shipping
9	Steven Markle	ST_CLERK	50	Stock Clerk	2008	5000	Shipping
10	Laura Bissot	ST_CLERK	50	Stock Clerk	2008	5000	Shipping
11	Mozhe Atkinson	ST_CLERK	50	Stock Clerk	2008	5000	Shipping
12	James Marlow	ST_CLERK	50	Stock Clerk	2008	5000	Shipping
13	Jason Mallin	ST_CLERK	50	Stock Clerk	2008	5000	Shipping
14	Michael Rogers	ST_CLERK	50	Stock Clerk	2008	5000	Shipping
15	Ki Gee	ST_CLERK	50	Stock Clerk	2008	5000	Shipping

Exercise SQL03-EX-05:

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

1. COMMIT

COMMIT in SQL is a transaction control language that is used to permanently save the changes done in the transaction in tables/databases. The database cannot regain its previous state after its execution of commit.

2. ROLLBACK

ROLLBACK in SQL is a transactional control language that is used to undo the transactions that have not been saved in the database. The command is only been used to undo changes since the last COMMIT.

	COMMIT	ROLLBACK
1.	COMMIT permanently saves the changes made by the current transaction.	ROLLBACK undo the changes made by the current transaction.
2.	The transaction can not undo changes after COMMIT execution.	Transaction reaches its previous state after ROLLBACK.
3.	When the transaction is successful, COMMIT is applied.	When the transaction is aborted, incorrect execution, system failure ROLLBACK occurs.
4.	COMMIT statement permanently save the state, when all the statements are executed successfully without any error.	In ROLLBACK statement if any operations fail during the completion of a transaction, it cannot permanently save the change and we can undo them using this statement.
5.	**Syntax of COMMIT statement are:** COMMIT;	**Syntax of ROLLBACK statement are:** ROLLBACK;