

Workbook

DATABASE MANAGEMENT SYSTEMS



Name	
Roll No.	
Batch	
Year	
Semester	

Department of Software Engineering
NED University of Engineering & Technology

TABLE OF CONTENTS

S.No	Experiment	Teacher's Sign
1	Installing Oracle SQL Plus	
2	Exploring SQL Plus Commands	
3	Restricting and Sorting Data	
4	SQL Functions	
5	Working with DATE in SQL Plus	
6	Conversion Functions	
7	Nesting Functions	
8	SQL Joins	
9	Introduction to No SQL Databases	
10	Installing MongoDB and Mongosh	
11	Working with MongoDB Shell “ Mongosh”	
12	Working with MongoDB GUI “Compass”	
13	Comparison, Logical Operators and Indexes in NoSQL Databases	
14	Open Ended Lab	

LAB No 2

Lab Exercise:

1. Create a table **Student** based on the following chart:

Column	Data type	Constraints
Student Id	Number (6)	Primary Key
Last Name	Varchar2(15)	Not NULL
First Name	Varchar2(15)	Not NULL
gender	Char(3)	

Confirm and validate the creation of the new table.

SOLUTION:

```
SQL> create table Student (Student_Id number(6) primary key, Last_Name varchar2(15) not null, First_Name varchar2(15) not null, Gender char(3));
Table created.
SQL> desc student;
Name                               Null?    Type
-----
STUDENT_ID                        NOT NULL NUMBER(6)
LAST_NAME                         NOT NULL VARCHAR2(15)
FIRST_NAME                       NOT NULL VARCHAR2(15)
GENDER                           CHAR(3)
```

2. Create a table **Dept** based on the following chart:

Column	Data type	Constraints
Dept Code	Char (3)	Not NULL
Dept Name	Varchar2(20)	Not NULL

Confirm and validate the creation of the new table.

SOLUTION:

```
SQL> create table Dept(Dept_Code char(3) not null, Dept_Name varchar2(20) not null);
Table created.
SQL> desc dept
Name                               Null?    Type
-----
DEPT_CODE                         NOT NULL CHAR(3)
DEPT_NAME                       NOT NULL VARCHAR2(20)
```

3. Add a new column **Location** to **Dept** table which has data type Char(7). Confirm and validate the modification of the table.

SOLUTION:

```
SQL> alter table Dept add(Location char(7));
Table altered.
SQL> desc dept;
Name                               Null?    Type
-----
DEPT_CODE                         NOT NULL CHAR(3)
DEPT_NAME                       NOT NULL VARCHAR2(20)
LOCATION                          CHAR(7)
```

4. Delete the column **Last_Name** from **Student**. Confirm and validate the modification of the table.

SOLUTION:

```
SQL> alter table Student drop column Last_Name;
Table altered.
SQL> desc Student;
Name Null? Type
-----
STUDENT_ID NOT NULL NUMBER(6)
FIRST_NAME NOT NULL VARCHAR2(15)
GENDER CHAR(3)
```

5. Increase **Last_Name** column to **25** characters long. Save the SQL statement as *ex5.sql*. Confirm and validate the modification of the table.

SOLUTION:

```
SQL> alter table Student add(Last_Name varchar2(15));
Table altered.
SQL> alter table student modify(Last_Name char(25) not null);
Table altered.
SQL> save ex5
Created file ex5.sql
SQL> desc student;
Name Null? Type
-----
STUDENT_ID NOT NULL NUMBER(6)
FIRST_NAME NOT NULL VARCHAR2(15)
GENDER CHAR(3)
LAST_NAME NOT NULL CHAR(25)
```

6. Create 2 more tables that you think are necessary for student data management.

SOLUTION:

```
GENDER CHAR(3)
LAST_NAME NOT NULL CHAR(25)

SQL> create table courses(Student_ID number(6) references student(Student_ID), Enrolled number(2) not null, BACKLOG NUMBER(2));
Table created.

SQL> create table personal_info(Student_ID number(6) references student(Student_ID), city varchar(15), college varchar(35));
Table created.

SQL> desc courses;
Name Null? Type
-----
STUDENT_ID NUMBER(6)
ENROLLED NOT NULL NUMBER(2)
BACKLOG NUMBER(2)

SQL> desc personal_info;
Name Null? Type
-----
STUDENT_ID NUMBER(6)
CITY VARCHAR2(15)
COLLEGE VARCHAR2(35)
```

7. Insert values in all the tables.

SOLUTION:

```
1 row created.

SQL> insert into student values(&student_id, '&first_name', '&gender', '&last_name');
Enter value for student_id: 25
Enter value for first_name: hamza
Enter value for gender: m
Enter value for last_name: narejo
old 1: insert into student values(&student_id, '&first_name', '&gender', '&last_name')
new 1: insert into student values(25, 'hamza', 'm', 'narejo')
1 row created.

SQL> insert into student values(&student_id, '&first_name', '&gender', '&last_name');
Enter value for student_id: 26
Enter value for first_name: uzma
Enter value for gender: f
Enter value for last_name: narejo
old 1: insert into student values(&student_id, '&first_name', '&gender', '&last_name')
new 1: insert into student values(26, 'uzma', 'f', 'narejo')
1 row created.

SQL> select *from student;

STUDENT_ID FIRST_NAME GEN LAST_NAME
-----
24 shehnila f narejo
25 hamza m narejo
26 uzma f narejo
```

```
SQL> insert into dept values('&Dept_Code', '&Dept_name', '&Location');
Enter value for dept_code: SE
Enter value for dept_name: Software eng
Enter value for location: karachi
old 1: insert into dept values('&Dept_Code', '&Dept_name', '&Location')
new 1: insert into dept values('SE', 'Software eng', 'karachi')
1 row created.

SQL> insert into dept values('&Dept_Code', '&Dept_name', '&Location');
Enter value for dept_code: se
Enter value for dept_name: software eng
Enter value for location: karachi
old 1: insert into dept values('&Dept_Code', '&Dept_name', '&Location')
new 1: insert into dept values('se', 'software eng', 'karachi')
1 row created.

SQL> select * from dept;

DEP DEPT_NAME LOCATIO
---
SE SOFTWARE ENG Karachi
SE Software eng karachi
se software eng karachi

SQL> _
```

```
SQL> insert into courses values(&Student_ID, &enrolled, &backlog);
Enter value for student_id: 24
Enter value for enrolled: 5
Enter value for backlog: 0
old 1: insert into courses values(&Student_ID, &enrolled, &backlog)
new 1: insert into courses values(24, 5, 0)
1 row created.

SQL> insert into courses values(&Student_ID, &enrolled, &backlog);
Enter value for student_id: 25
Enter value for enrolled: 6
Enter value for backlog: 0
old 1: insert into courses values(&Student_ID, &enrolled, &backlog)
new 1: insert into courses values(25, 6, 0)
1 row created.

SQL> insert into courses values(&Student_ID, &enrolled, &backlog);
Enter value for student_id: 26
Enter value for enrolled: 6
Enter value for backlog: 1
old 1: insert into courses values(&Student_ID, &enrolled, &backlog)
new 1: insert into courses values(26, 6, 1)
1 row created.
```

```
SQL> insert into personal_info values(&Student_ID, '&city', '&college');
Enter value for student_id: 24
Enter value for city: karachi
Enter value for college: aitchison
old 1: insert into personal_info values(&Student_ID, '&city', '&college')
new 1: insert into personal_info values(24, 'karachi', 'aitchison')
1 row created.

SQL> insert into personal_info values(&Student_ID, '&city', '&college');
Enter value for student_id: 25
Enter value for city: karachi
Enter value for college: adamjee
old 1: insert into personal_info values(&Student_ID, '&city', '&college')
new 1: insert into personal_info values(25, 'karachi', 'adamjee')
1 row created.

SQL> insert into personal_info values(&Student_ID, '&city', '&college');
Enter value for student_id: 26
Enter value for city: karachi
Enter value for college: adamjee
old 1: insert into personal_info values(&Student_ID, '&city', '&college')
new 1: insert into personal_info values(26, 'karachi', 'adamjee')
1 row created.
```

8. Select specific data from the tables

SOLUTION:

```
SQL> select * from student where Student_ID=25;

STUDENT_ID FIRST_NAME GEN LAST_NAME
-----
25 hamza m narejo

SQL> select * from dept where location='Karachi';

DEP DEPT_NAME LOCATIO
---
SE SOFTWARE ENG Karachi

SQL> _
```

9. Select entire data from the tables

SOLUTION:

```
SQL> select * from student;

STUDENT_ID FIRST_NAME GEN LAST_NAME
-----
24 shehnila f narejo
25 hamza m narejo
26 uzma f narejo

SQL> select * from dept;

DEP DEPT_NAME LOCATIO
---
SE SOFTWARE ENG Karachi
SE Software eng karachi
se software eng karachi

SQL> select * from courses;

STUDENT_ID ENROLLED BACKLOG
-----
24 5 0
25 6 0
26 6 1

SQL> select * from personal_info;

STUDENT_ID CITY COLLEGE
-----
24 karachi aitchison
25 karachi adamjee
26 karachi adamjee
```

10. Truncate student table

SOLUTION:

```
SQL> truncate table student;
truncate table student
*
ERROR at line 1:
ORA-02266: unique/primary keys in table referenced by enabled foreign keys
```

11. Add 2 new columns to dept table

SOLUTION:

```
SQL> alter table dept add(Dept_No number(4), Seats number(5));
Table altered.

SQL> desc dept;
Name Null? Type
-----
DEPT_CODE NOT NULL CHAR(3)
DEPT_NAME NOT NULL VARCHAR2(20)
LOCATION CHAR(7)
DEPT_NO NUMBER(4)
SEATS NUMBER(5)
```

12. Delete a particular column from dept table.

SOLUTION:

```
SQL> alter table dept drop column dept_no;
Table altered.

SQL> desc dept;
Name Null? Type
-----
DEPT_CODE NOT NULL CHAR(3)
DEPT_NAME NOT NULL VARCHAR2(20)
LOCATION CHAR(7)
SEATS NUMBER(5)
```

13. Create arithmetic calculation.

SOLUTION:

```
SQL> select student_id, first_name, student_id+100 from student;

STUDENT_ID FIRST_NAME      STUDENT_ID+100
-----
          24 shehnila          124
          25 hamza            125
          26 uzma             126
```

14. Select distinct values from the column

SOLUTION:

```
SQL> select * from student where student_id=24;

STUDENT_ID FIRST_NAME      GEN LAST_NAME
-----
          24 shehnila          f   narejo

SQL> select * from courses where enrolled=5;

STUDENT_ID  ENROLLED  BACKLOG
-----
          24          5          0
```

15. Concatenate columns and display

SOLUTION:

```
SQL> select first_name||last_name as Full_Name from student;

FULL_NAME
-----
shehnilanarejo
hamzanarejo
uzmanarejo
```

16. Drop all tables.

SOLUTION:

```
SQL> drop table courses;
Table dropped.

SQL> drop table personal_info;
Table dropped.

SQL> drop table dept;
Table dropped.

SQL> drop table students;
drop table students
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> drop table student;
Table dropped.

SQL> _
```

LAB No 3

Lab Exercise:

1. Create a query to display the name and salary of employees earning more than \$4000.

SOLUTION:

```
SQL> Run SQL Command Line
SQL> select emp_name, salary from employee_Info where salary >4000;

EMP_NAME                                SALARY
-----                                -
Shehnila                                5000
```

2. Create a query to display the employee name and department number for employee number 7839.

SOLUTION:

```
SQL> select emp_name, dept_no from employee_Info where emp_id ='7839';

EMP_NAME                                DEPT_NO
-----                                -
Shehnila                                10
```

3. Modify lab to display the name and salary for all employees whose salary is not in the range of \$5,000 and \$12,000.

SOLUTION:

```
SQL> select emp_name, salary from employee_Info where salary NOT BETWEEN 5000 AND 12000;

EMP_NAME                                SALARY
-----                                -
JONES                                   2975
BLAKE                                   2850
CLARK                                   2450
SCOTT                                   3000
FORD                                    3000
Hamza                                   800
TURNER                                  1500
ADAMS                                   1100
JAMES                                   950
MILLER                                  1300

10 rows selected.
```

4. Display the employee name, job , and hiredate of employees hired between February 20, 1998, and May 1, 1998. Order the query in ascending order by start date

SOLUTION:

```
SQL> SELECT emp_name, job, hire_date FROM employee_Info WHERE hire_date BETWEEN TO_DATE('1998-02-20', 'YYYY-MM-DD') AND TO_DATE('1998-05-01', 'YYYY-MM-DD');

EMP_NAME                                HIRE_DATE
-----                                -
SMITH                                    15-MAR-98
CLERK
```


5. Display the name and department number of all employees in departments 20 and 30 in alphabetical order by name.

SOLUTION:

```
SQL> SELECT emp_name, dept_no FROM employee_Info WHERE dept_no IN ('20', '30') ORDER BY emp_name ASC;
```

EMP_NAME	DEPT_NO
ADAMS	20
BLAKE	30
FORD	20
Hamza	20
JAMES	30
JONES	20
SCOTT	20
SMITH	20
TURNER	30

9 rows selected.

6. Modify lab to list the name and salary of employees who earn between \$5,000 and \$12,000, and are in department 20 or 50. Label the columns Employee and Monthly Salary , respectively.

SOLUTION:

```
SQL> SELECT * FROM employee_Info WHERE salary BETWEEN 2000 AND 4000 AND dept_no IN ('10', '30');
```

EMP_ID	EMP_NAME	JOB	COMMISSION	DEPT_NO	HIRE_DATE	SALARY
7698	BLAKE	MANAGER		30	11-JUN-92	2850
7782	CLARK	MANAGER		10	14-MAY-93	2450

7. Display the last name and hire date of every employee who was hired in 1994.

SOLUTION:

```
SQL> SELECT emp_name, hire_date FROM employee_Info WHERE EXTRACT(YEAR FROM hire_date) = 1994;
```

EMP_NAME	HIRE_DATE
John Doe	15-JUN-94

8. Display the last name and job title of all employees who do not have a manager.

SOLUTION:

```
SQL> SELECT emp_name AS "Last Name", job AS "Job Title" FROM employee_Info WHERE mgr_id IS NULL;

Last Name
-----
Job Title
-----
Shehnaz
PRESIDENT
JONES
MANAGER
BLAKE
MANAGER

Last Name
-----
Job Title
-----
CLARK
MANAGER
SCOTT
ANALYST
FORD
ANALYST

Last Name
-----
Job Title
-----
Hamza
CLERK
TURNER
```

```
Last Name
-----
Job Title
-----
JAMES
CLERK

MILLER
CLERK

SMITH
CLERK

Last Name
-----
Job Title
-----
John Doe
Analyst

khan
Analyst

14 rows selected.
SQL>
```

9. Display the last name, salary, and commission for all employees who earn commissions. Sort data in descending order of salary and commissions.

SOLUTION:

```
SQL> SELECT emp_name, salary, commission FROM employee_Info WHERE commission IS NOT NULL ORDER BY salary DESC, commission DESC;

EMP_NAME                                SALARY COMMISSION
-----
TURNER                                  1500      0

SQL>
```

10. Display the last names of all employees where the third letter of the name is an a.

SOLUTION:

```
SQL> SELECT emp_name FROM employee_Info WHERE SUBSTR(emp_name, 3, 1) = 'A';

EMP_NAME
-----
BLAKE
CLARK
ADAMS
```

11. Display the last name of all employees who have an a and an e in their name.

SOLUTION:

```
SQL> SELECT emp_name FROM employee_Info WHERE emp_name LIKE '%A%E%';

EMP_NAME
-----
BLAKE
JAMES
```

12. Display the last name, job, and salary for all employees whose job is salesman or clerk and whose salary is not equal to \$2,500, \$3,500, or \$800.

SOLUTION:

```
SQL> SELECT emp_name, job, salary FROM employee_Info WHERE (job = 'SALESMAN' OR job = 'CLERK') AND salary NOT IN (2500, 3500, 800);

EMP_NAME
-----
JOB
-----
SALARY
-----
TURNER
SALESMAN
1500

ADAMS
CLERK
1100

JAMES
CLERK
950

EMP_NAME
-----
JOB
-----
SALARY
-----
MILLER
CLERK
1300

SMITH
CLERK
1200
```

LAB No 4

Lab Exercise

A) CASE MANIPULATION FUNCTIONS

1. Upper Function :

```
SQL> select upper(ename)from employee;

UPPER(ENAME)
-----
SHEHNILA
HAMZA
WAQAR
UZMA
USMAN
MEERA
AZAM
```

2. Lower Function:

```
SQL> select lower(ename)from employee;

LOWER(ENAME)
-----
shehnila
hamza
waqar
uzma
usman
meera
azam
```

B) CHARACTER MANIPULATION FUNCTION

1. SQL> select ename, lower(ename),length(ename), INSTR(ename,'s'),CONCAT(deptno,ename),RPAD(ename,7,'*') FROM employee WHERE upper(deptno) = 'se';

```
SQL> SELECT ename, lower(ename), length(ename), INSTR(ename, 's'), CONCAT(deptno, ename), RPAD(ename, 7, '*')
2 FROM employee;

ENAME          LOWER(ENAME)    LENGTH(ENAME)  INSTR(ENAME, 'S')
CONCAT(DEPTNO,ENAME)  RPAD(ENAME,7,'*')
-----
shehnila      shehnila        8              1
10shehnila
hamza         hamza           5              0
Shamza
waqar         waqar           5              0
7waqar
uzma          uzma            4              0
Buzma
usman         usman           5              2
Busman
meera         meera           5              0
9meera
azam          azam            4              0
10azam
7 rows selected.
SQL>
```

INSTR(string, substring, [start_position], [nth_appearance])

```
SQL> SELECT ENAME, INSTR(DEPTNO, 'i', 3, 2) FROM EMPLOYEE WHERE ENAME = 'shehnila';
```

ENAME	INSTR(DEPTNO, 'I', 3, 2)
shehnila	0

```
SQL> SELECT ENAME, INSTR(DEPTNO, 'i', 2, 1) FROM EMPLOYEE WHERE ENAME = 'shehnila';
```

ENAME	INSTR(DEPTNO, 'I', 2, 1)
shehnila	0

However, the start position and nth appearance must be aligned otherwise the answers to query would be difficult to comprehend

```
SQL> SELECT ENAME, INSTR(ENAME, 'a', 3, 2) FROM EMPLOYEE WHERE DEPTNO = '5';
```

ENAME	INSTR(ENAME, 'A', 3, 2)
hamza	0

```
SQL> SELECT ENAME, INSTR(ENAME, 'A', 3, 1) FROM EMPLOYEE WHERE DEPTNO = '5';
```

ENAME	INSTR(ENAME, 'A', 3, 1)
hamza	0

Too many arguments may also exhaust the query such as

```
SQL> select ename, instr(ename, 'a', 1, 1, 2) from employee where deptno='csiti';
select ename, instr(ename, 'a', 1, 1, 2) from employee where deptno='csiti'
*
ERROR at line 1:
ORA-00939: too many arguments for function
```

2. Using LPAD and RPAD:

```
SQL>select          ename,lower(ename),length(ename),
INSTR(ename,'s'),CONCAT(deptno,ename),LPAD(ename,7,'*')
FROM employee WHERE upper(deptno) = 'se';
```

```
SQL> SELECT ENAME, LOWER(ENAME), LENGTH(ENAME), INSTR(ENAME, 's'), CONCAT(DEPTNO, ENAME), LPAD(ENAME, 7, '*') FROM EMPLOYEE WHERE LOWER(DEPTNO) = '10';
```

ENAME	LOWER(ENAME)	LENGTH(ENAME)	INSTR(ENAME, 'S')
shehnila	shehnila	8	1
10shehnila	shehnil		
azam	azam	4	0
10azam	***azam		

3. SUBSTR:

```
SQL> select ename, deptno, substr(deptno, 1, 2) from employee;
```

ENAME	DEPTNO	SUBSTR(D
shehnila	10	10
hamza	5	5
waqar	7	7
uzma	3	3
usman	8	8
meera	9	9
azam	10	10

7 rows selected.

4. Any string can also be subtracted:

SQL> select substr(deptno,1,2) from employee;

```
SQL> select ename, substr(deptno,1,2)from employee;
ENAME                               SUBSTR(D
-----
shehnila                            10
hamza                               5
waqar                               7
uzma                                3
usman                               8
meera                               9
azam                                10
7 rows selected.

SQL> select ename, substr(deptno,2,1) from employee;
ENAME                               SUBS
-----
shehnila                            0
hamza                               0
waqar                               0
uzma                                0
usman                               0
meera                               0
azam                                0
7 rows selected.

SQL> _
```

5. Number Functions:

SQL> select round(45.923) from dual;

```
SQL> select round(45,923)from dual;
ROUND(45,923)
-----
45
```

SQL> select round(45.123) from dual;

```
SQL> select round(45,123)from dual;
ROUND(45,123)
-----
45

SQL> _
```

SQL> select round(45.923,2) from dual;

```
SQL> select round(45.123,2)from dual;
ROUND(45.123,2)
-----
45.12

SQL> _
```

We can also round off negative numbers

```
SQL> select round(-45.923,2)from dual;
ROUND(-45.923,2)
-----
-45.92
```

Rounding off values with negative parameters

SQL> select round(45.923,-1) from dual;

```
SQL> select round(45.923,-1) from dual;

ROUND(45.923,-1)
-----
                50
```

```
SQL> select round(44.923,-1) from dual;

ROUND(44.923,-1)
-----
                40
```

Rounding off to nearest 100

```
SQL> select round(45.923,-2) from dual;

ROUND(45.923,-2)
-----
                0
```

```
SQL> select round(55.923,-2) from dual;

ROUND(55.923,-2)
-----
               100
```

Using the TRUNC Function

```
SQL> select round(45.923) from dual;

ROUND(45.923)
-----
                46
```

SQL> select trunc(45.923,1) from dual;

SQL> select trunc(45.923,2) from dual;

```
SQL> select trunc(45.923,1) from dual;

TRUNC(45.923,1)
-----
               45.9
```

```
SQL> select trunc(45.923,2) from dual;

TRUNC(45.923,2)
-----
               45.92
```

Truncating a date to a specific precision:

SQL>select TRUNC(SYSDATE, 'MONTH') FROM dual;

```
SQL> select TRUNC(SYSDATE, 'MONTH') FROM dual;

TRUNC(SYS
-----
01-MAY-24
```

SQL>select TRUNC(SYSDATE, 'YEAR') FROM dual;

SQL>select TRUNC(SYSDATE, 'DAY') FROM dual;

```
SQL> select TRUNC(SYSDATE, 'YEAR') FROM dual;

TRUNC(SYS
-----
01-JAN-24

SQL> select TRUNC(SYSDATE, 'DAY') FROM dual;

TRUNC(SYS
-----
26-MAY-24
```

Using the MOD Function

SQL> select MOD(10,3) FROM employee;

```
SQL> select MOD(10,3) FROM employee;

MOD(10,3)
-----
         1
         1
         1
         1
         1
         1
         1
         1

7 rows selected.
```

SQL> select MOD(salary, 500) FROM employee;

```
SQL> select MOD(salary, 500) FROM employee;

MOD(SALARY,500)
-----
         0
         0
        400
         0
         0
         0
         0

7 rows selected.
```


LAB No 5

Lab Exercise:

SQL> select sysdate from dual;

SOLUTION:

```
SQL> select sysdate from dual;
SYSDATE
-----
27-MAY-24
```

TO_CHAR function to convert the SYSDATE to a string format with the desired date and time representation.

SOLUTION:

```
SQL> select TO_CHAR(sysdate, 'DD-MM-YYY HH24:MI:SS') from dual;
TO_CHAR(SYSDATE, 'D
-----
27-05-024 13:56:12
```

- ROUND(SYSDATE, 'MONTH')
- ROUND(SYSDATE, 'YEAR')
- TRUNC(SYSDATE, 'MONTH')
- TRUNC(SYSDATE, 'YEAR')

SOLUTION:

```
SQL> select round(sysdate, 'month') from dual;
ROUND(SYS
-----
01-JUN-24

SQL> select round(sysdate, 'year') from dual;
ROUND(SYS
-----
01-JAN-24

SQL> select trunc(sysdate, 'month') from dual;
TRUNC(SYS
-----
01-MAY-24

SQL> select trunc(sysdate, 'year') from dual;
TRUNC(SYS
-----
01-JAN-24

SQL> _
```

Working with the dates in table employee

SOLUTION:

```
SQL> alter table employee add hire_date date;
alter table employee add hire_date date
*
ERROR at line 1:
ORA-01430: column being added already exists in table

SQL> desc employee
  Name                                                    Null?         Type
-----
EMPID                                                    NOT NULL     VARCHAR2(10)
ENAME                                                    NOT NULL     VARCHAR2(20)
SALARY                                                    NOT NULL     NUMBER(10)
DEPTNO                                                    NOT NULL     VARCHAR2(10)
HIRE_DATE                                                DATE
```

```
SQL> UPDATE employee SET hire_date = TO_DATE('2023-06-13', 'YYYY-MM-DD') WHERE empid = 1;
1 row updated.

SQL> select * from employee;

EMPID      ENAME            SALARY  DEPTNO      HIRE_DATE
-----
1          shehnila         5000    10          13-JUN-23
2          hamza            5000    5           13-JUN-23
3          waqar            400     7
4          uzma             5500    3
5          usman            6000    8
6          meera            7000    9
7          azam             33000   10

7 rows selected.

SQL> UPDATE employee SET hire_date = TO_DATE('13-06-2023', 'DD-MM-YYYY') WHERE empid = 2;
1 row updated.
```

Rounding and truncating the hire_date;

SOLUTION:

```
SQL> select round(hire_date, 'month') from employee;

ROUND(HIR
-----
01-JUN-23
01-JUN-23

7 rows selected.

SQL> select round(hire_date, 'year')from employee;

ROUND(HIR
-----
01-JAN-23
01-JAN-23

7 rows selected.

SQL>
```

Arithmetic Operations with dates:

SQL> select ename, (SYSDATE-hire_date)/7 AS WEEKS FROM employee;

SOLUTION:

```
SQL> select ename, (SYSDATE-hire_date)/7 AS WEEKS FROM employee;
ENAME                               WEEKS
-----
shehnila                            49.9442146
hamza                               49.9442146
waqar
uzma
usman
meera
azam
7 rows selected.
```

Adding a number of days:

SQL> select hire_date+ 7 FROM employee;

SOLUTION:

```
SQL> select hire_date+ 7 FROM employee;
HIRE_DATE
-----
20-JUN-23
20-JUN-23
7 rows selected.
```

Subtracting a number of days:

SQL> select hire_date-6 FROM employee;

SOLUTION:

```
SQL> select hire_date-6 FROM employee;
HIRE_DATE
-----
07-JUN-23
07-JUN-23
7 rows selected.
```

Adding months to date;

SQL> select add_months(hire_date,6) FROM employee;

SOLUTION:

```
SQL> select add_months(hire_date,6) FROM employee;

ADD_MONTH
-----
13-DEC-23
13-DEC-23

7 rows selected.
```

Subtracting a number of months:

SQL> select add_months(hire_date,-6) FROM employee;

SOLUTION:

```
SQL> select add_months(hire_date,-6) FROM employee;

ADD_MONTH
-----
13-DEC-22
13-DEC-22

7 rows selected.
```

Adding a number of years:

SQL>select add_months(hire_date,6*12) FROM employee;

SOLUTION:

```
SQL> select add_months(hire_date,6*12) FROM employee;

ADD_MONTH
-----
13-JUN-29
13-JUN-29

7 rows selected.
```

Subtracting a number of years:

SQL> select add_months(hire_date,-6*12) FROM employee;

SOLUTION:

```
SQL> select add_months(hire_date,-6*12) FROM employee;

ADD_MONTH
-----
13-JUN-17
13-JUN-17

7 rows selected.
```

LAB No 6

Lab Exercise:

Example of implicit data type conversion

SQL> select emp_id, emp_name, emp_salary * 1.1 AS increased_salary FROM employees;

SOLUTION:

```
SQL> select empid, ename, salary * 1.1 AS increased_salary FROM employee;
```

EMPID	ENAME	INCREASED_SALARY
1	shehnila	5500
2	hamza	5500
3	waqar	440
4	uzma	6050
5	usman	6600
6	meera	7700
7	azam	36300

7 rows selected.

Example of implicit data type conversion

You can also alter the existing emp_salary table

SQL> alter table emp_pp modify emp_salary varchar2(10);

SOLUTION:

```
SQL> ALTER TABLE employee ADD (new_salary VARCHAR2(10));
Table altered.
SQL> UPDATE employee SET new_salary = TO_CHAR(salary);
7 rows updated.
SQL> ALTER TABLE employee DROP COLUMN salary;
Table altered.
SQL> ALTER TABLE employee RENAME COLUMN new_salary TO salary;
Table altered.
SQL> SELECT * FROM employee;
```

EMPID	ENAME	DEPTNO	HIRE_DATE	SALARY
1	shehnila	10		5000
2	hamza	5		5000
3	waqar	7		400
4	uzma	3		5500
5	usman	8		6000
6	meera	9		7000
7	azam	10		33000

7 rows selected.

SQL>

Now add values to this column

SOLUTION:

```
SQL> UPDATE employee SET salary = NULL WHERE ename = 'shehnila';
1 row updated.

SQL> alter table employee modify salary varchar2(10);
Table altered.

SQL> update employee set salary='1000' where ename='hamza';
1 row updated.

SQL> update employee set salary='2000' where ename='waqar';
1 row updated.
```

```
SQL> select * from employee;

EMPID      ENAME      DEPTNO      HIRE_DATE      SALARY
-----
1          shehnila      10
2          hamza         5             1000
3          waqar         7             2000
4          uzma          3             5500
5          usman         8             6000
6          meera         9             7000
7          azam          10            33000

7 rows selected.

SQL> _
```

Converting a date to a string with a specific format

```
SQL> select TO_CHAR(sysdate, DD-MON-YYYY HH:MI:SS) FROM dual;
```

SOLUTION:

```
SQL> SELECT TO_CHAR(sysdate, 'DD-MON-YYYY HH:MI:SS') FROM dual;
TO_CHAR(SYSDATE, 'DD-MON-YYYYH
-----
29-MAY-2024 05:00:09

SQL> SELECT TO_CHAR(sysdate, 'DD-MON-YYYY HH:MI:SS') FROM dual;
TO_CHAR(SYSDATE, 'DD-MON-YYYYH
-----
29-MAY-2024 05:00:11

SQL> SELECT TO_CHAR(sysdate, 'DD-MON-YYYY HH:MI:SS') FROM dual;
TO_CHAR(SYSDATE, 'DD-MON-YYYYH
-----
29-MAY-2024 05:00:15

SQL> _
```

You can use different formats and visualize the output.

SOLUTION:

```
SQL> select TO_CHAR(sysdate, 'DD-Mo-YY HH:MI')from dual;  
select TO_CHAR(sysdate, 'DD-Mo-YY HH:MI')from dual  
      *  
ERROR at line 1:  
ORA-01821: date format not recognized  
  
SQL> select TO_CHAR(sysdate, 'D-MO-YY HH:MI')from dual;  
      select TO_CHAR(sysdate, 'D-MO-YY HH:MI')from dual  
      *  
ERROR at line 1:  
ORA-01821: date format not recognized  
  
SQL> select TO_CHAR(sysdate, 'D-MO-YYY HH:MI')from dual;  
      select TO_CHAR(sysdate, 'D-MO-YYY HH:MI')from dual  
      *  
ERROR at line 1:  
ORA-01821: date format not recognized
```

Formatting a number with decimal places and thousands separator

SQL>select TO_CHAR(12345.6789, 9,999.99) FROM dual;

SOLUTION:

```
SQL> select TO_CHAR(12345.6789, '9,999.99') FROM dual;  
  
TO_CHAR(1  
-----  
#####  
  
SQL> select TO_CHAR(12345.6789, '99,999.99') FROM dual;  
  
TO_CHAR(12  
-----  
12,345.68  
  
SQL>
```

Converting a string to a number

SQL>select TO_NUMBER(12345.67) FROM dual;

SOLUTION:

```
SQL> select TO_NUMBER(12345.67) FROM dual;  
  
TO_NUMBER(12345.67)  
-----  
12345.67  
  
SQL>
```


TO_DATE:

Converting a string to a date

SQL>select TO_DATE(2023-07-04, 'YYYY-MM-DD') FROM dual;

SOLUTION:

```
SQL> SELECT TO_DATE('2023-07-04', 'YYYY-MM-DD') FROM dual;

TO_DATE( '
-----
04-JUL-23
```

TTO_TIMESTAMP:

TO_TIMESTAMP(string, [format])

Converting a string to a timestamp

SQL> select TO_TIMESTAMP(2023-07-04 12:34:56,';YYYY-MM-DD HH24:MI:SS') FROM
dual;

SOLUTION:

```
SQL> SELECT TO_TIMESTAMP('2023-07-04 12:34:56', 'YYYY-MM-DD HH24:MI:SS') FROM dual;

TO_TIMESTAMP( '2023-07-0412:34:56', 'YYYY-MM-DDHH24:MI:SS' )
-----
04-JUL-23 12.34.56.000000000 PM
SQL> _
```

CAST

Casting a value to a different data type

SQL> select CAST(123 AS NUMBER) FROM dual;

SOLUTION:

```
SQL> select CAST('123' AS NUMBER) FROM dual;

CAST( '123'ASNUMBER)
-----
123
```

LAB No 7

Lab Exercise

Example of nesting functions

SQL> select UPPER(SUBSTR('Hello, World!', 1, 5)) FROM dual;

```
SQL> select UPPER(SUBSTR('Hello, World!', 1, 5)) FROM dual;
UPPER
-----
HELLO
```

Nested functions to calculate the square of a number and round it to the nearest integer

SQL> select ROUND(SQRT(25)) FROM dual;

```
SQL> select round(sqrt(25)) from dual;
ROUND(SQRT(25))
-----
5
```

NVL(expression, substitute_value)

Using NVL to handle NULL values

SQL> select NVL(emp_salary, 0) FROM employees;

```
SQL> SELECT NVL(SALARY, '0') FROM employee;
NVL(SALARY
-----
0
1000
2000
5500
6000
7000
33000
7 rows selected.
SQL> SELECT NVL(SALARY, '2') FROM employee;
NVL(SALARY
-----
2
1000
2000
5500
6000
7000
33000
7 rows selected.
```

Using NVL with a substitute value from another column

SQL> select NVL(emp_salary, emp_name) FROM employees;

```
SQL> select NVL(salary, ename) FROM employee;
NVL(SALARY,ENAME)
-----
shehnila
1000
2000
5500
6000
7000
33000
7 rows selected.
SQL>
```

NVL2(expression, value_if_not_null, value_if_null)

Using NVL2 to handle NULL values with different substitute values

SQL> select NVL2(emp_salary, Salary is not null, Salary is null;) FROM empp;

```
SQL> select NVL2(salary, 'salary is not null','salary is null')from employee;
NVL2(SALARY,'SALAR
-----
salary is null
salary is not null
salary is not null
salary is not null
salary is not null
salary is not null
salary is not null
7 rows selected.
```

Using NVL2 with different substitute values from other columns

SQL> select NVL2(salary, bonus, commission) FROM employees;

```
SQL> select NVL2(salary, ename, empid) from employee;
NVL2(SALARY,ENAME,EM
-----
1
hamza
waqar
uzma
usman
meera
azam
7 rows selected.
SQL> _
```

Using DECODE with a default value

SQL> select employee_name, DECODE(department_id, 10, Finance, 20, Marketing, Other Department;) AS department FROM employees;

```
SQL> select ename,decode(deptno,'ME',salary,890)as dept from employee;
ENAME                               DEPT
-----
shehnila                            890
hamza                               890
waqar                               890
uzma                                890
usman                               890
meera                               890
azam                                890
7 rows selected.
```

LAB No 8

Lab Exercise

1. Equijoin:

SQL> select column1, column2 FROM table1 JOIN table2, ON table1.column_name = table2.column_name;

```
SQL> desc employee
Name                                     Null?      Type
-----
EMPID                                   NOT NULL   VARCHAR2(10)
ENAME                                   NOT NULL   VARCHAR2(20)
DEPTNO                                  VARCHAR2(10)
HIRE_DATE                              DATE
SALARY                                  VARCHAR2(10)

SQL> select * from employee;

EMPID      ENAME      DEPTNO      HIRE_DATE      SALARY
-----
1          shehnala    10
2          hamza       5              1000
3          waqar       7              2000
4          uzma        3              5500
5          usman       8              6000
6          meera       9              7000
7          azam        10             33000

7 rows selected.
```

```
SQL> desc student
Name                                     Null?      Type
-----
STDID                                   VARCHAR2(10)
SNAME                                   VARCHAR2(15)
SCOURSE                                VARCHAR2(10)
SYEAR                                  VARCHAR2(12)

SQL> select * from student;

STDID      SNAME      SCOURSE      SYEAR
-----
2          amar       math          2023
3          john       science       2023
4          lisa       history       2024
5          mike       english       2022
6          sara       biology       2023
7          paul       chemistry     2024

6 rows selected.

SQL> _
```

Joining Tables with WHERE Clause

```
SQL> SELECT *FROM employee, student WHERE employee.empid = student.stdid;

EMPID      ENAME      DEPTNO      HIRE_DATE      SALARY      STDID
-----
2          hamza      5              1000           2
3          waqar      7              2000           3
4          uzma       3              5500           4
5          usman      8              6000           5
6          meera      9              7000           6
7          azam       10             33000          7

6 rows selected.

SQL>
```

Joining tables using JOIN Keyword

```
SQL> SELECT employee.empid, employee.ename FROM employee JOIN student ON employee.empid = student.stdid;
```

EMPID	ENAME
2	hamza
3	waqar
4	uzma
5	usman
6	meera
7	azam

6 rows selected.

2. Non equi join:

SQL> select column1, column2, FROM table1 JOIN table2 ON table1.column_name < table2.column_name;

```
SQL> SELECT employee.empid, employee.ename FROM employee JOIN student ON employee.empid < student.stdid;
```

EMPID	ENAME
1	shehnila
1	shehnila
2	hamza
1	shehnila
2	hamza
3	waqar
1	shehnila
2	hamza
3	waqar
4	uzma
1	shehnila

EMPID	ENAME
2	hamza
3	waqar
4	uzma
5	usman
1	shehnila
2	hamza
3	waqar
4	uzma
5	usman
6	meera

21 rows selected.

```
SQL> SELECT employee.empid, employee.ename FROM employee JOIN student ON employee.empid > student.stdid;
```

EMPID	ENAME
3	waqar
4	uzma
5	usman
6	meera
7	azam
4	uzma
5	usman
6	meera
7	azam
5	usman
6	meera

EMPID	ENAME
7	azam
6	meera
7	azam
7	azam

15 rows selected.

SQL> _

3. Cross join

SQL>select * FROM table1 CROSS JOIN table2;

```
SQL> select * FROM employee CROSS JOIN student;
```

EMPID	ENAME	DEPTNO	HIRE_DATE	SALARY	STDID
1	amar	10			2
2	amar	5		1000	2
3	amar	7		2000	2

EMPID	ENAME	DEPTNO	HIRE_DATE	SALARY	STDID
1	amar	10			2
2	amar	5		1000	2
3	amar	7		2000	2

4. Natural join

SQL> select * FROM table1 NATURAL JOIN table2;

```
SQL> select empid , ename from employee NATURAL JOIN student;
```

EMPID	ENAME
1	shehnila
2	hamza
3	waqar
4	uzma
5	usman
6	meera
7	azam
1	shehnila
2	hamza
3	waqar
4	uzma

5. Inner join

SQL>select * FROM table1 INNER JOIN table2 ON table1.column_name = Table2.column_name;

```
SQL> SELECT employee.empid, employee.ename FROM employee INNER JOIN student ON employee.empid = student.stdid;
```

EMPID	ENAME
2	hamza
3	waqar
4	uzma
5	usman
6	meera
7	azam

6 rows selected.

```
SQL>
```

6. Outer join

1) Left outer join

SQL> select * FROM table1 LEFT OUTER JOIN table2 ON table1.column_name =table2.column_name;

```
SQL> SELECT employee.empid, employee.ename FROM employee LEFT OUTER JOIN student ON employee.empid = student.stdid;
```

EMPID	ENAME
2	hamza
3	waqar
4	uzma
5	usman
6	meera
7	azam
1	shehnila

7 rows selected.

```
SQL> _
```

2)Right outer join

**SQL> select * FROM table1 RIGHT OUTER JOIN table2
ON table1.column_name= table2.column_name;**

```
SQL> SELECT employee.empid, employee.ename FROM employee RIGHT OUTER JOIN student ON employee.empid = student.stdid;
```

EMPID	ENAME
2	hamza
3	waqar
4	uzma
5	usman
6	meera
7	azam

6 rows selected.

```
SQL>
```

3)Full outer join:

**SQL>select * FROM table1 FULL OUTER JOIN table2 ON
table1.column_name =table2.column_name;**

```
SQL> SELECT employee.empid, employee.ename FROM employee FULL OUTER JOIN student ON employee.empid = student.stidid;

EMPID      ENAME
-----
1          shehnaila
2          hamza
3          waqar
4          uzma
5          usman
6          meera
7          azam
7 rows selected.

SQL>
```

LAB No 11

Lab Exercise:

```
test> show dbs
admin      40.00 KiB
config    108.00 KiB
local      72.00 KiB
test> use admin
switched to db admin
admin> use school
switched to db school
school> show dbs
admin      40.00 KiB
config    108.00 KiB
local      72.00 KiB
```

```
school> db.createCollection("student")
{ ok: 1 }
school> show dbs
admin      40.00 KiB
config    108.00 KiB
local      72.00 KiB
school     8.00 KiB
school> db.dropDatabase()
{ ok: 1, dropped: 'school' }
school> db.student.insertOne({name:"shehnila",age:30, gpa:3.0})
school>
  acknowledged: true,
school>
}
school>
```

```
school> db.student.find()
[
  {
    _id: ObjectId('6687159ffc0b1e100790defe'),
    name: 'shehnila',
    age: 30,
    gpa: 3
  }
]
school> db.student.insertM_
```



```

school> db.students.insertMany([ {name: "usama", age: 13, gpa: 3.0}, {name: "uzair", age: 17, gpa: 5.3}, {name: "ali",
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64b186e647ff296b5d4a7d4f"),
    '1': ObjectId("64b186e647ff296b5d4a7d50"),
    '2': ObjectId("64b186e647ff296b5d4a7d51")
  }
}
}
school> db.students.find()
[
  {
    _id: ObjectId("64b1865e47ff296b5d4a7d4d"),
    name: 'moin',
    age: 18,
    gpa: 3.6
  },
  {
    _id: ObjectId("64b1867947ff296b5d4a7d4e"),
    name: 'moiz',
    age: 20,
    gpa: 3.8
  },
  {
    _id: ObjectId("64b186e647ff296b5d4a7d4f"),
    name: 'usama',
    age: 13,
    gpa: 3
  }
]

```

```

school> db.students.find().sort({ name: 1 })
[
  { _id: ObjectId("64b1885247ff296b5d4a7d53"), age: 48 },
  { _id: ObjectId("64b1886747ff296b5d4a7d54"), gpa: 4 },
  { _id: ObjectId("64b1887147ff296b5d4a7d55"), fulltime: false },
  {
    _id: ObjectId("64b1888147ff296b5d4a7d56"),
    registerDate: ISODate("1970-01-01T00:00:00.000Z")
  },
  { _id: ObjectId("64b1888947ff296b5d4a7d57"), graduationDate: null },
  {
    _id: ObjectId("64b188a747ff296b5d4a7d58"),
    courses: [ 'Math', 'AP', 'ARW' ]
  },
  {
    _id: ObjectId("64b188c147ff296b5d4a7d59"),
    address: { street: 'mainK.', city: 'Karachi', zip: 6657 }
  },
  {
    _id: ObjectId("64b186e647ff296b5d4a7d51"),
    name: 'ali',
    age: 13,
    gpa: 2.9
  },
  {
    _id: ObjectId("64b1865e47ff296b5d4a7d4d"),
    name: 'moin',
    age: 18,
    gpa: 3.6
  }
]

```

```

school> db.student.insertOne({name: "Ayesha", age: 32, gpa: 2.8, fullTime: false, registerDate: new Date(), graduationDate: null, courses: ["Datastructure", "oop", "Calculus"], address: {street: "123 ned university", city: "Karachi", zip: 23456}})
{
  acknowledged: true,
  insertedId: ObjectId("64c17364fdd332ea482f9bc2")
}
school>

```

```

school> db.student.insertOne({name: "Ayesha", age: 32, gpa: 2.8, fullTime: false, registerDate: new Date(), graduationDate: null, courses: ["Datastructure", "oop", "Calculus"], address: {street: "123 ned university", city: "Karachi", zip: 23456}})
{
  acknowledged: true,
  insertedId: ObjectId("64c17364fdd332ea482f9bc2")
}
school>

```

```

school> db.student.find().sort({name:1})
[
  {
    _id: ObjectId("64c17364fdd332ea482f9bc2"),
    name: 'Ayesha',
    age: 32,
    gpa: 2.8,
    fullTime: false,
    registerDate: ISODate("2023-07-26T19:26:28.369Z"),
    graduationDate: null,
    courses: [ 'Datastructure', 'oop', 'Calculus' ],
    address: { street: '123 ned university', city: 'Karachi', zip: 23456 }
  },
  {
    _id: ObjectId("64c16e50fdd332ea482f9bbf"),
    name: 'amna',
    age: 38,
    gpa: 1.5
  },
  {
    _id: ObjectId("64c16e50fdd332ea482f9bc0"),
    name: 'hina',
    age: 27,
    gpa: 4
  },
  {
    _id: ObjectId("64c16dc2fdd332ea482f9bbe"),

```

```

school> db.student.find().sort({age:1})
[
  {
    _id: ObjectId("64c16e50fdd332ea482f9bc1"),
    name: 'simra',
    age: 18,
    gpa: 2.5
  },
  {
    _id: ObjectId("64c16e50fdd332ea482f9bc0"),
    name: 'hina',
    age: 27,
    gpa: 4
  },
  {
    _id: ObjectId("64c16dc2fdd332ea482f9bbe"),
    name: 'shiza',
    age: 30,
    gpa: 3.2
  },

```

```

school> db.student.find().sort({name:-1})
[
  {
    _id: ObjectId("64c16e50fdd332ea482f9bc1"),
    name: 'simra',
    age: 18,
    gpa: 2.5
  },
  {
    _id: ObjectId("64c16dc2fdd332ea482f9bbe"),
    name: 'shiza',
    age: 30,
    gpa: 3.2
  },
  {
    _id: ObjectId("64c16e50fdd332ea482f9bc0"),
    name: 'hina',
    age: 27,
    gpa: 4
  },
]

```

```

school> db.student.findOne()
{
  _id: ObjectId('6687159ffc0b1e100790defe'),
  name: 'shehnila',
  age: 30,
  gpa: 3
}
school>

school> db.student.find({name:"shehnila"})
[
  {
    _id: ObjectId('6687159ffc0b1e100790defe'),
    name: 'shehnila',
    age: 30,
    gpa: 3
  }
]
school> _

```

```

school> db.student.find({gpa: 3.2, fullTime: false})

school> db.student.find({gpa: 2.8, fullTime: false})
[
  {
    _id: ObjectId("64c17364fdd332ea482f9bc2"),
    name: 'Ayesha',
    age: 32,
    gpa: 2.8,
    fullTime: false,
    registerDate: ISODate("2023-07-26T19:26:28.369Z"),
    graduationdate: null,
    courses: [ 'Datastructure', 'oop', 'Calculus' ],
    address: { street: '123 ned university', city: 'Karachi', zip: 23456 }
  }
]
school>

```

```
school> db.student.deleteMany({fullTime:false})
{ acknowledged: true, deletedCount: 1 }
school>
```

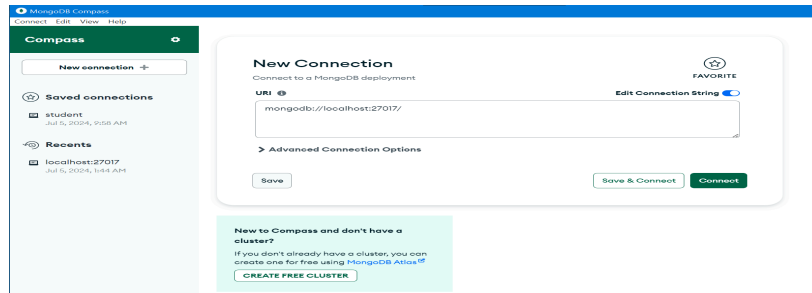
```
school> db.student.find()
[
  {
    _id: ObjectId("64c16e50fdd332ea482f9bbf"),
    name: 'amna',
    age: 38,
    gpa: 1.5,
    fullTime: true
  },
  {
    _id: ObjectId("64c16e50fdd332ea482f9bc0"),
    name: 'hina',
    age: 27,
    gpa: 4,
    fullTime: true
  },
  {
    _id: ObjectId("64c16e50fdd332ea482f9bc1"),
    name: 'simra',
    age: 18,
    gpa: 2.5,
    fullTime: true
  }
]
school>
```

```
school> db.student.deleteMany({registerDate:{$exists: false}})
{ acknowledged: true, deletedCount: 3 }
school>
```

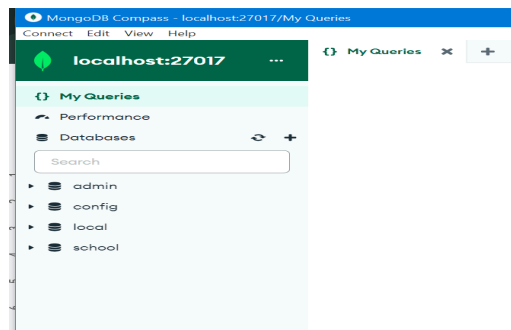
LAB No 12

Lab Exercise:

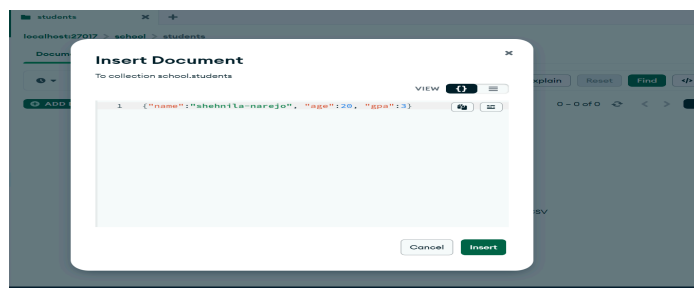
Syntax of Command for connection



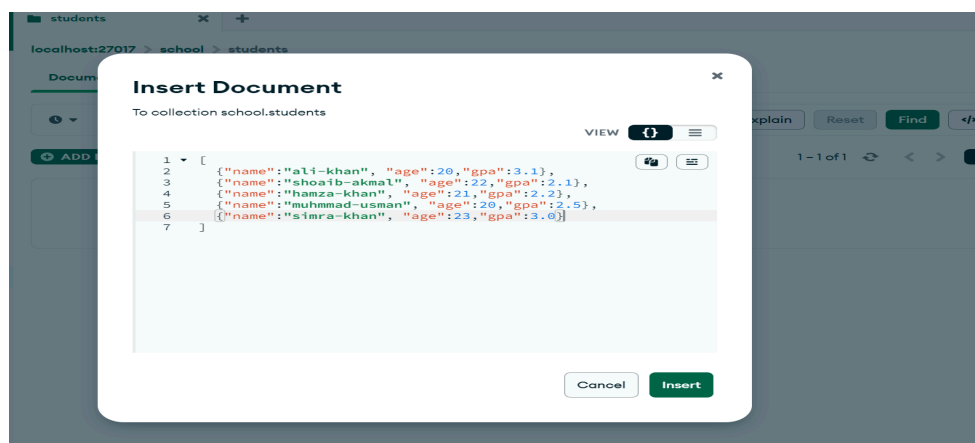
To view all databases in MongoDB Compass:



To insert One Data at a time:



To insert more than one documents:



Finding a record/document:

Documents 6 Aggregations Schema Indexes 1 Validation

Type a query: { name: "shehnila-narejo" } or [Generate query](#) ⚡

[Explain](#) [Reset](#) [Find](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

1 - 1 of 1

```
{
  "_id": ObjectId("668bcea877bc6ec093be13e0"),
  "name": "shehnila-narejo",
  "age": 20,
  "gpa": 3
}
```

Sorting data:

Documents 6 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#) ⚡

[Explain](#) [Reset](#) [Find](#) [Options](#)

Project { field: 0 }

Sort { name: 1 }

Collation { locale: 'simple' }

Index Hint { field: -1 }

Max Time MS 60000

Skip 0 Limit 0

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

1 - 6 of 6

```
{
  "_id": ObjectId("668bcf6077bc6ec093be13e2"),
  "name": "ali-khan",
  "age": 20,
  "gpa": 3.1
}
```

```
{
  "_id": ObjectId("668bcf6077bc6ec093be13e4"),
  "name": "hamza-khan",
  "age": 21,
  "gpa": 2.2
}
```

```
{
  "_id": ObjectId("668bcf6077bc6ec093be13e5")
}
```

Limit the record:

Documents 6 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#) ⚡

[Explain](#) [Reset](#) [Find](#) [Options](#)

Project { field: 0 }

Sort { name: 1 }

Collation { locale: 'simple' }

Index Hint { field: -1 }

Max Time MS 60000

Skip 0 Limit 2

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

1 - 2 of 2

```
{
  "_id": ObjectId("668bcf6077bc6ec093be13e2"),
  "name": "ali-khan",
  "age": 20,
  "gpa": 3.1
}
```

```
{
  "_id": ObjectId("668bcf6077bc6ec093be13e4"),
  "name": "hamza-khan",
  "age": 21,
  "gpa": 2.2
}
```

Optimizing the query:

localhost:27017

students

localhost:27017 > school > students

Documents 6 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Project {name: true}

Sort { field: -1 } or [['field', -1]]

Collation { locale: 'simple' }

Index Hint { field: -1 }

Max Time MS 60000

Skip 0 Limit 0

EXPORT DATA

1 - 6 of 6

```

{ "_id": ObjectId("668bcea877bc6ec093be13e0"), "name": "shehnika-narejo" }
{ "_id": ObjectId("668bcbf6077bc6ec093be13e2"), "name": "ali-khan" }
{ "_id": ObjectId("668bcbf6077bc6ec093be13e3"), "name": "shoaib-akmal" }
{ "_id": ObjectId("668bcbf6077bc6ec093be13e4"), "name": "hamza-khan" }

```

Here we are still able to view the Object ID as it is unique for all records. If we do not want to see the Object ID, then we can further optimize the query. {name: true, _id: false}

localhost:27017

students

localhost:27017 > school > students

Documents 6 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Project {name: true, _id: false}

Sort { field: -1 } or [['field', -1]]

Collation { locale: 'simple' }

Index Hint { field: -1 }

Max Time MS 60000

Skip 0 Limit 0

EXPORT DATA

1 - 6 of 6

```

{ "name": "shehnika-narejo" }
{ "name": "ali-khan" }
{ "name": "shoaib-akmal" }
{ "name": "hamza-khan" }

```

Updating a document:

localhost:27017 ...

My Queries
Performance
Databases
admin
config
local
school
employees
students

students

localhost:27017 > school > students

Documents 6 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

EXPLAIN Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

1 - 6 of 6

Document modified. CANCEL UPDATE

```

1  _id: ObjectId('668bcea877bc6ec093be13e0')
2  name: "shehnila-narejo"
3  age: 20
   gpa: 35

```

ObjectId
String
Int32
Int32

```

_id: ObjectId('668bcf6077bc6ec093be13e2')
name: "ali-khan"
age: 20
gpa: 3.1

```

```

_id: ObjectId('668bcf6077bc6ec093be13e3')
name: "shoaib-akmal"
age: 22
gpa: 2.1

```

Similarly we can add a new field by clicking on the + icon besides the trash icon and then click on update. This will add a new data to our document.

ADD DATA EXPORT DATA UPDATE DELETE

1 - 6 of 6

Document modified. CANCEL UPDATE

```

_id: ObjectId('668bcea877bc6ec093be13e0')
name: "shehnila-narejo"
age: 20
gpa: 35
work: "fulltime"

```

```

_id: ObjectId('668bcf6077bc6ec093be13e2')
name: "ali-khan"
age: 20
gpa: 3.1

```

```

_id: ObjectId('668bcf6077bc6ec093be13e3')
name: "shoaib-akmal"

```

The fullTime field has been added to the document with the name “shehnila”

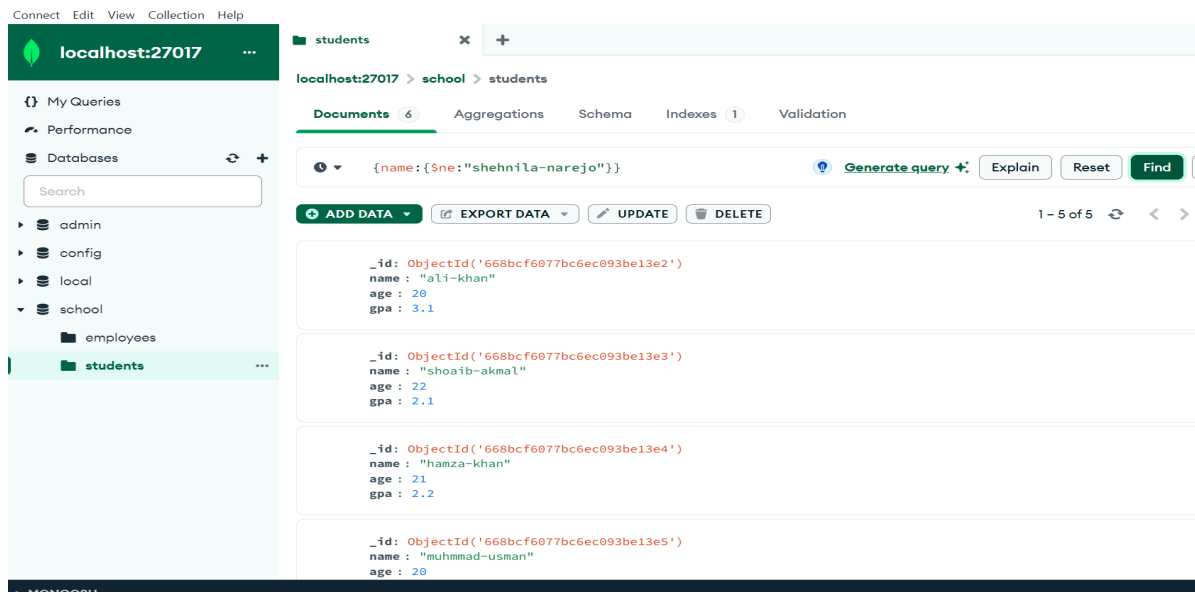
Lab 13

Find all names except one name

```
mongosh mongodb://127.0.0.1:27017/mongosh?directConnection=true&serverSelectionTimeoutMS=30000
school> db.students.find({name:{$ne:"shehnila-narejo"}})
[
  {
    _id: ObjectId('668bcf6077bc6ec093be13e2'),
    name: 'ali-khan',
    age: 20,
    gpa: 3.1
  },
  {
    _id: ObjectId('668bcf6077bc6ec093be13e3'),
    name: 'shoaib-akmal',
    age: 22,
    gpa: 2.1
  },
  {
    _id: ObjectId('668bcf6077bc6ec093be13e4'),
    name: 'hamza-khan',
    age: 21,
    gpa: 2.2
  },
  {
    _id: ObjectId('668bcf6077bc6ec093be13e5'),
    name: 'muhammad-usman',
    age: 20,
    gpa: 2.5
  },
  {
    _id: ObjectId('668bcf6077bc6ec093be13e6'),
    name: 'simra-khan',
    age: 23,
    gpa: 3
  }
]
school>
```

In MongoDB Compass

The Not Equal != Comparison operator:



It will show all records except for those that contain “shehnila-narejo” in the name field.

Less than equal to Operator:

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure with 'school' selected and 'students' expanded. The main panel shows a query: `{ age: { $lte: 20 }}`. The results tab displays three documents:

```
{ "_id": "ObjectId('668bcea877bc6ec093be13e0')", "name": "shehnika-narejo", "age": 20, "gpa": 3.5, "work": "fulltime" }, { "_id": "ObjectId('668bcf6077bc6ec093be13e2')", "name": "ali-khan", "age": 20, "gpa": 3.1 }, { "_id": "ObjectId('668bcf6077bc6ec093be13e5')", "name": "muhammad-usman", "age": 20, "gpa": 2.5 }
```

Greater than operator:

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure with 'school' selected and 'students' expanded. The main panel shows a query: `{ age: { $gt: 20 }}`. The results tab displays three documents:

```
{ "_id": "ObjectId('668bcf6077bc6ec093be13e3')", "name": "shoaib-akmal", "age": 22, "gpa": 2.1 }, { "_id": "ObjectId('668bcf6077bc6ec093be13e4')", "name": "hamza-khan", "age": 21, "gpa": 2.2 }, { "_id": "ObjectId('668bcf6077bc6ec093be13e6')", "name": "sitora-khan", "age": 23, "gpa": 3 }
```

Less than equal to operator

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure with 'school' selected and 'students' expanded. The main panel shows a query: `{ age: { $gte: 20 }}`. The results tab displays four documents:

```
{ "_id": "ObjectId('668bcea877bc6ec093be13e0')", "name": "shehnika-narejo", "age": 20, "gpa": 3.5, "work": "fulltime" }, { "_id": "ObjectId('668bcf6077bc6ec093be13e2')", "name": "ali-khan", "age": 20, "gpa": 3.1 }, { "_id": "ObjectId('668bcf6077bc6ec093be13e3')", "name": "shoaib-akmal", "age": 22, "gpa": 2.1 }, { "_id": "ObjectId('668bcf6077bc6ec093be13e4')", "name": "hamza-khan" }
```

In Between Operator:

The screenshot shows the MongoDB Compass interface. On the left, the 'Databases' sidebar is open, showing a collection named 'students'. The main panel displays a query: `{ gpa: { $gte: 1, $lte: 3 } }`. The results show four documents with GPA values 2.1, 2.2, 2.5, and 2.3, all of which are between 1 and 3.

```
{ gpa: { $gte: 1, $lte: 3 } }
```

_id	name	age	gpa
ObjectId('668bcf6077bc6ec093be13e3')	shoaib-akmal	22	2.1
ObjectId('668bcf6077bc6ec093be13e4')	hamza-khan	21	2.2
ObjectId('668bcf6077bc6ec093be13e5')	muhmmad-usman	20	2.5
ObjectId('668bcf6077bc6ec093be13e6')	simra-khan	23	2.3

In Operator:

The screenshot shows the MongoDB Compass interface. The query is `{ name: { $in: ["shehnila-narejo", "simra-khan"] } }`. The results show two documents: one with name 'shehnila-narejo' and age 20, and another with name 'simra-khan' and age 23.

```
{ name: { $in: ["shehnila-narejo", "simra-khan"] } }
```

_id	name	age	gpa	work
ObjectId('668bcea877bc6ec093be13e0')	shehnila-narejo	20	35	fulltime
ObjectId('668bcf6077bc6ec093be13e6')	simra-khan	23	3	

Not In operator:

The screenshot shows the MongoDB Compass interface. The query is `{ "name": { "$nin": ["shehnila-narejo", "ali-khan", ""] } }`. The results show three documents with names 'shoaib-akmal', 'hamza-khan', and 'muhmmad-usman', none of which are in the excluded list.

```
{ "name": { "$nin": ["shehnila-narejo", "ali-khan", ""] } }
```

_id	name	age	gpa
ObjectId('668bcf6077bc6ec093be13e3')	shoaib-akmal	22	2.1
ObjectId('668bcf6077bc6ec093be13e4')	hamza-khan	21	2.2
ObjectId('668bcf6077bc6ec093be13e5')	muhmmad-usman	20	2.5

Logical Operators In Mongosh

AND operator

```
School> db.student.find({$and: [{name: "simra"}, {age: {$lte:22}}]})
[
  {
    _id: ObjectId("64c2930706989d266393fa5d"),
    name: 'simra',
    age: 1.2
  }
]
```

OR operator

```
School> db.student.find({$or: [{name: true}, {age: {$lte:22}}]})
[
  {
    _id: ObjectId("64c2930706989d266393fa5c"),
    name: 'hina',
    age: 20,
    gpa: 2.2
  },
  {
    _id: ObjectId("64c2930706989d266393fa5d"),
    name: 'simra',
    age: 1.2
  }
]
```

NOR operator

```
School> db.student.find({$nor: [{name: true}, {age: {$lte:22}}]})
[
  {
    _id: ObjectId("64c2918906989d266393fa5a"),
    name: 'shiza',
    age: 30,
    gpa: 3.2,
    fullTime: 'true'
  },
  {
    _id: ObjectId("64c2930706989d266393fa5e"),
    name: 'maria',
    age: 25,
    gpa: 4
  },
  {
    _id: ObjectId("64c296f706989d266393fa60"),
    name: 'ayesha',
    age: 32,
    gpa: 2.8,
    fullTime: false,
    graduationdate: null,
    courses: [ 'datastructures', 'oop', 'calculus' ],
    address: { street: 'neduniversity', city: 'karachi', zip: 23456 }
  }
]
```

NOT operator

```
School> db.student.find({age: {$not: {$gte:30}}})
[
  {
    _id: ObjectId("64c2930706989d266393fa5c"),
    name: 'hina',
    age: 20,
    gpa: 2.2
  },
  {
    _id: ObjectId("64c2930706989d266393fa5d"),
    name: 'simra',
    age: 1.2
  },
  {
    _id: ObjectId("64c2930706989d266393fa5e"),
    name: 'maria',
    age: 25,
    gpa: 4
  }
]
```

In MongoDB Compass AND operator

Documents 6 Aggregations Schema Indexes 1 Validation

🕒 ⌵ `{ $and: [{ name: "shehnila-narejo" }, { age: { $lte: 30 } }] }` 🧠 Generate query ⚡ Explain Reset

➕ ADD DATA ▾ 📄 EXPORT DATA ▾ ✎ UPDATE 🗑 DELETE 1 - 1 of 1 ↺

```

_id: ObjectId('668bcea877bc6ec093be13e0')
name: "shehnila-narejo"
age: 20
gpa: 35
work: "fulltime"

```

OR operator

Documents 6 Aggregations Schema Indexes 1 Validation

🕒 ⌵ `{ $or: [{ name: "shehnila-narejo" }, { age: { $lte: 30 } }] }` 🧠 Generate query ⚡ Explain Re

➕ ADD DATA ▾ 📄 EXPORT DATA ▾ ✎ UPDATE 🗑 DELETE 1 - 6 of 6

```

_id: ObjectId('668bcea877bc6ec093be13e0')
name: "shehnila-narejo"
age: 20
gpa: 35
work: "fulltime"

```

```

_id: ObjectId('668bcf6077bc6ec093be13e2')
name: "ali-khan"
age: 20
gpa: 3.1

```

```

_id: ObjectId('668bcf6077bc6ec093be13e3')
name: "shoaib-akmal"
age: 22
gpa: 2.1

```

```

_id: ObjectId('668bcf6077bc6ec093be13e4')

```

NOR operator

🕒 ⌵ `{ $nor: [{ name: "shehnila-narejo" }, { age: { $lte: 20 } }] }` 🧠 Generate query ⚡ Explain Reset F

➕ ADD DATA ▾ 📄 EXPORT DATA ▾ ✎ UPDATE 🗑 DELETE 1 - 3 of 3 ↺

```

_id: ObjectId('668bcf6077bc6ec093be13e3')
name: "shoaib-akmal"
age: 22
gpa: 2.1

```

```

_id: ObjectId('668bcf6077bc6ec093be13e4')
name: "hamza-khan"
age: 21
gpa: 2.2

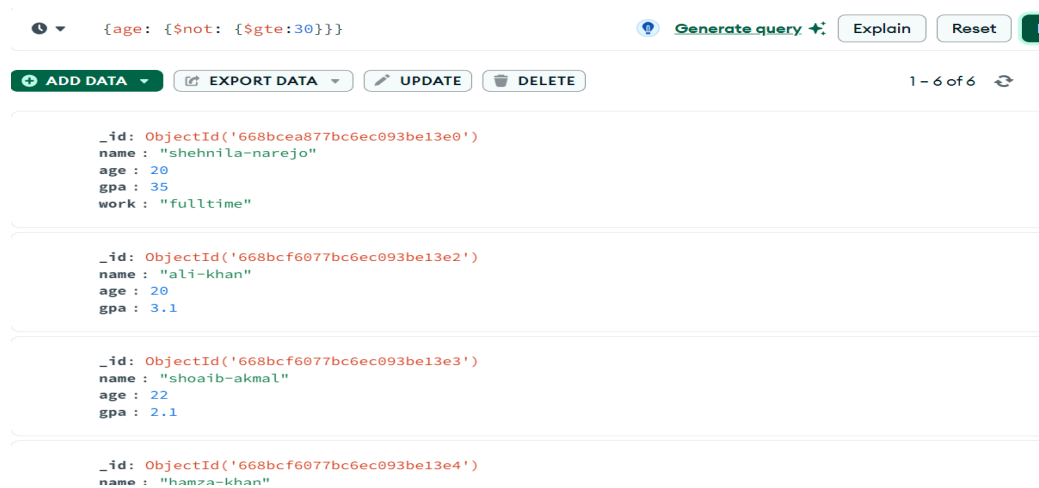
```

```

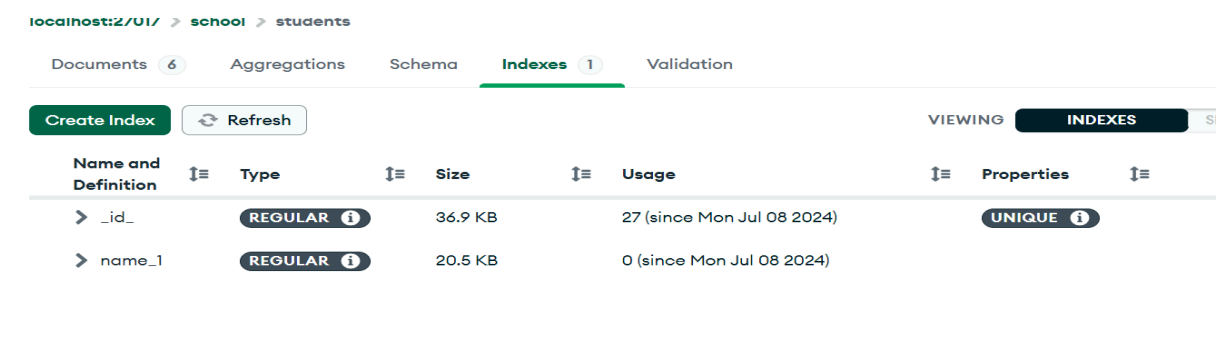
_id: ObjectId('668bcf6077bc6ec093be13e6')
name: "simra-khan"
age: 23
gpa: 3

```

NOT operator



Creating an Index:



To drop an Index:

