

Assignment-2

Q.1 Create a table employee having following columns:

firstname, lastname, title, age and salary

The screenshot shows the Oracle Database Express Edition interface. The title bar says "ORACLE Database Express Edition". The user is "MOHD NASIR". The menu bar has "Home", "Logout", and "Help". The main area shows the SQL Commands page. A SQL statement is being typed into the editor:

```
create table employee(firstname varchar(20),lastname varchar(20), title varchar(20), age number(3), salary number(7))
```

Buttons for "Save" and "Run" are visible. Below the editor, tabs for "Results", "Explain", "Describe", "Saved SQL", and "History" are shown. The status bar at the bottom says "Table created."

Description Of Table.

The screenshot shows the Oracle Database Express Edition interface. The title bar says "ORACLE Database Express Edition". The user is "MOHD NASIR". The menu bar has "Home", "Logout", and "Help". The main area shows the SQL Commands page. A SQL statement is being typed into the editor:

```
describe employee
```

Buttons for "Save" and "Run" are visible. Below the editor, tabs for "Results", "Explain", "Describe", "Saved SQL", and "History" are shown. The status bar at the bottom says "Object Type TABLE Object EMPLOYEE". A detailed table of columns is displayed:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	FIRSTNAME	Varchar2	20	-	-	-	✓	-	-
	LASTNAME	Varchar2	20	-	-	-	✓	-	-
	TITLE	Varchar2	20	-	-	-	✓	-	-
	AGE	Number	-	3	0	-	✓	-	-
	SALARY	Number	-	7	0	-	✓	-	-

Page number "1 - 5" is at the bottom right of the table.

Q.2 Enter the data of eight employee into employee table.

The screenshot shows the Oracle Database Express Edition interface. The title bar says "ORACLE Database Express Edition". The user is "MOHD NASIR". The menu bar has "Home", "Logout", and "Help". The main area shows the SQL Commands page. A SQL statement is being typed into the editor:

```
insert all
into employee values ('Mohd', 'Nasir', 'Programmer', 19,35000)
into employee values ('Ayusman', 'Joshi', 'UI Designer', 21,500000)
into employee values ('Deepak', 'Kumar', 'Developer', 21,43000)
into employee values ('Naushad', 'Zakir', 'AI Engineer', 21, 40000)
into employee values ('Anirudh', 'Chandan', 'Programmer', 22,40000)
into employee values ('Vaibhav', 'Pratap', 'Data Science', 22,700000)
into employee values ('Jagtar', 'Singh', 'Developer', 22,55000)
into employee values ('Aakash', 'Pundeer', 'Programmer', 22, 45000)
select* from dual
```

Buttons for "Save" and "Run" are visible. Below the editor, tabs for "Results", "Explain", "Describe", "Saved SQL", and "History" are shown. The status bar at the bottom says "8 row(s) inserted."

Q.3 On the basis of employee table answer the following questions:

(i) Select all columns for everyone in your emp table.

Autocommit Display 10 Save

```
select *  
from employee
```

Results Explain Describe Saved SQL History

FIRSTNAME	LASTNAME	TITLE	AGE	SALARY
Mohd	Nasir	Programmer	19	35000
Ayusman	Joshi	UX Designer	21	500000
Deepak	Kumar	Developer	21	43000
Naushad	Zakir	AI Engineer	21	40000
Anirudh	Chandan	Programmer	22	40000
Vaibhav	Pratap	Data Science	22	700000
Jagtar	Singh	Developer	22	55000
Aakash	Pundeer	Programmer	22	45000

8 rows returned in 0.00 seconds [CSV Export](#)

(ii) Select first and last names for everybody that's under 30 year old.

Autocommit Display 10 Save

```
select firstname, lastname  
from employee  
where age<30
```

Results Explain Describe Saved SQL History

FIRSTNAME	LASTNAME
Mohd	Nasir
Ayusman	Joshi
Deepak	Kumar
Naushad	Zakir
Anirudh	Chandan
Vaibhav	Pratap
Jagtar	Singh
Aakash	Pundeer

8 rows returned in 0.00 seconds [CSV Export](#)

(iii) Select all columns for everyone with a salary over 45000.

Autocommit Display 10 Save

```
select *  
from employee  
where salary>45000
```

Results Explain Describe Saved SQL History

FIRSTNAME	LASTNAME	TITLE	AGE	SALARY
Ayusman	Joshi	UX Designer	21	500000
Vaibhav	Pratap	Data Science	22	700000
Jagtar	Singh	Developer	22	55000

3 rows returned in 0.00 seconds [CSV Export](#)

(iv) Select firstname, lastname and salary for anyone with 'Programmer' in their title.

Autocommit Display 10 Save

```
select firstname, lastname, salary  
from employee  
where title='Programmer'
```

Results Explain Describe Saved SQL History

FIRSTNAME	LASTNAME	SALARY
Mohd	Nasir	35000
Anirudh	Chandan	40000
Aakash	Pundeer	45000

3 rows returned in 0.00 seconds [CSV Export](#)

(v) Select the firstname for everyone whose lastname contains 'a'.

The screenshot shows a MySQL query interface. The SQL query is:

```
select firstname  
from employee  
where lastname like '%a%'
```

The results table has one column labeled "FIRSTNAME". The data is:

FIRSTNAME
Mohd
Deepak
Naushad
Anirudh
Vaibhav

Below the table, it says "5 rows returned in 0.00 seconds" and there is a "CSV Export" link.

(vi) select all columns for everyone whose firstname starts with 'D'.

The screenshot shows a MySQL query interface. The SQL query is:

```
select *  
from employee  
where firstname like 'D%'
```

The results table has five columns: FIRSTNAME, LASTNAME, TITLE, AGE, and SALARY. The data is:

FIRSTNAME	LASTNAME	TITLE	AGE	SALARY
Deepak	Kumar	Developer	21	43000

Below the table, it says "1 rows returned in 0.00 seconds" and there is a "CSV Export" link.

ASSIGNMENT-3

1. select all record of employees.

Autocommit Display 10

```
select* from hr.employees
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	-	103	60
105	David	Austin	DAUSTIN	590.423.4569	25-JUN-97	IT_PROG	4800	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-98	IT_PROG	4800	-	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	4200	-	103	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-94	FI_MGR	12000	-	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	16-AUG-94	FI_ACCOUNT	9000	-	108	100

More than 10 rows available. Increase rows selector to view more rows.

2. Select all records of employees where salary is >10000

Autocommit Display 10

```
select* from hr.employees
where salary>10000
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	-	100	90
108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-94	FI_MGR	12000	-	101	100
114	Den	Raphaely	DRAPHEAL	515.127.4561	07-DEC-94	PU_MAN	11000	-	100	30
145	John	Russell	JRUSSEL	011.44.1344.429268	01-OCT-96	SA_MAN	14000	.4	100	80
146	Karen	Partners	KPARTNER	011.44.1344.467268	05-JAN-97	SA_MAN	13500	.3	100	80
147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	10-MAR-97	SA_MAN	12000	.3	100	80
148	Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	15-OCT-99	SA_MAN	11000	.3	100	80
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-00	SA_MAN	10500	.2	100	80

More than 10 rows available. Increase rows selector to view more rows.

3. Select first_name, last_name and title of employees having salary between 25000

>=salary >=10000

Autocommit Display 10

```
select first_name, last_name
from hr.employees
where salary between 10000 and 25000
```

Results Explain Describe Saved SQL History

FIRST_NAME	LAST_NAME
Steven	King
Neena	Kochhar
Lex	De Haan
Nancy	Greenberg
Den	Raphaely
John	Russell
Karen	Partners

4. select first_name, emp_id , phone no of records of employees having department id=90

Autocommit Display 10 Save

```
select first_name, employee_id, phone_number
from hr.employees
where department_id=90
```

Results Explain Describe Saved SQL History

FIRST_NAME	EMPLOYEE_ID	PHONE_NUMBER
Steven	100	515.123.4567
Neena	101	515.123.4568
Lex	102	515.123.4569

3 rows returned in 0.00 seconds [CSV Export](#)

5. Select all records of employees in the increasing order of salary.

Autocommit Display 10 Save

```
select *
from hr.employees
order by salary
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
132	TJ	Olson	TJOLSON	650.124.8234	10-APR-99	ST_CLERK	2100	-	121	50
128	Steven	Markle	SMARKLE	650.124.1434	08-MAR-00	ST_CLERK	2200	-	120	50
136	Hazel	Philtanker	HPHILTAN	650.127.1634	06-FEB-00	ST_CLERK	2200	-	122	50
127	James	Landry	JLANDRY	650.124.1334	14-JAN-99	ST_CLERK	2400	-	120	50
135	Ki	Gee	KGEE	650.127.1734	12-DEC-99	ST_CLERK	2400	-	122	50
119	Karen	Colmenares	KCOLMENA	515.127.4566	10-AUG-99	PU_CLERK	2500	-	114	30
131	James	Marlow	JAMRLOW	650.124.7234	16-FEB-97	ST_CLERK	2500	-	121	50
140	Joshua	Patel	JPATEL	650.121.1834	06-APR-98	ST_CLERK	2500	-	123	50
144	Peter	Vargas	PVARGAS	650.121.2004	09-JUL-98	ST_CLERK	2500	-	124	50
182	Martha	Sullivan	MSULLIVA	650.507.9878	21-JUN-99	SH_CLERK	2500	-	120	50

More than 10 rows available. Increase rows selector to view more rows.

6. Select all records of employees in the decreasing order of salary.

Autocommit Display 10 Save

```
select *
from hr.employees
order by salary desc
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	-	100	90
145	John	Russell	JRUSSEL	011.44.1344.429268	01-OCT-96	SA_MAN	14000	4	100	80
146	Karen	Partners	KPARTNER	011.44.1344.467268	05-JAN-97	SA_MAN	13500	3	100	80
201	Michael	Harstein	MHARSTE	515.123.5555	17-FEB-98	MK_MAN	13000	-	100	20
108	Nancy	Greenberg	NGREENBE	515.123.4569	17-AUG-94	FL_MGR	12000	-	101	100
147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	10-MAR-97	SA_MAN	12000	3	100	80
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	12000	-	101	110
168	Lisa	Ozer	LOZER	011.44.1343.929268	11-MAR-97	SA_REP	11500	25	148	80

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds [CSV Export](#)

7. Select first_name of employees in the increasing order of salary and if salary matches then decreasing order of the department_id.

Autocommit Display 100 Save

```
select *
from hr.employees
order by salary , department_id desc
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
131	James	Marlow	JAMRLOW	650.124.7234	16-FEB-97	ST_CLERK	2500	-	121	50
191	Randall	Perkins	RPERKINS	650.505.4876	19-DEC-99	SH_CLERK	2500	-	122	50
144	Peter	Vargas	PVARGAS	650.121.2004	09-JUL-98	ST_CLERK	2500	-	124	50
182	Martha	Sullivan	MSULLIVA	650.507.9878	21-JUN-99	SH_CLERK	2500	-	120	50
119	Karen	Colmenares	KCOLMENA	515.127.4566	10-AUG-99	PU_CLERK	2500	-	114	30
143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98	ST_CLERK	2600	-	124	50
199	Douglas	Grant	DGRANT	650.507.9844	13-JAN-00	SH_CLERK	2600	-	124	50
198	Donald	O'Connell	DOCONNEL	650.507.9833	21-JUN-99	SH_CLERK	2600	-	124	50
118	Guy	Himuro	GHIMURO	515.127.4565	15-NOV-98	PU_CLERK	2600	-	114	30
139	John	Seo	JSEO	650.121.2019	12-FEB-98	ST_CLERK	2700	-	123	50
126	Irene	Mikkilineni	IMIKKIL	650.124.1224	28-SEP-98	ST_CLERK	2700	-	120	50
130	Mozhe	Atkinson	MATKINSO	650.124.6234	30-OCT-97	ST_CLERK	2800	-	121	50

8. Retrieve first_name and department_id and the salary of all the purchase managers and display them with modified salary(5% increase in salaryrename the column by new_salary) for there good work.

```
Autocommit Display 10 Save Run
select first_name,department_id, (105/100 * salary) as "new salary"
from hr.employees
```

FIRST_NAME	DEPARTMENT_ID	New Salary
Steven	90	25200
Neena	90	17850
Lex	90	17850
Alexander	60	9450
Bruce	60	6300
David	60	5040
Valli	60	5040
Diana	60	4410
Nancy	100	12600
Daniel	100	9450

More than 10 rows available. Increase rows selector to view more rows.

9. Retrieve first_name of all programmers of IT department whose first name starts with a and ends with b.

```
Autocommit Display 500 Save Run
select first_name
from hr.employees e, hr.departments d
where e.department_id=d.department_id and department_name='IT' and first_name like 'a%b'
```

FIRST_NAME
no data found

10. Retrieve first_name of all programmers of IT department whose first name contains a substring 'ish'.

```
Autocommit Display 10 Save Run
select first_name
from hr.employees e, hr.departments d
where e.department_id=d.department_id and department_name='IT' and first_name like '%ish%'
```

FIRST_NAME
no data found

11. Retrieve all the records except David, Peter,Oliver,Allan.

```
Autocommit Display 10 Save Run
select *
from hr.employees e
where first_name !='David' and first_name !='Peter'
and first_name !='Oliver' and first_name !='Allan'
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	-	-	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-88	IT_PROG	4800	-	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	4200	-	103	60
108	Nancy	Greenberg	NGREENBE	515.124.4568	17-AUG-94	FI_MGR	12000	-	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	16-AUG-94	FI_ACCOUNT	9000	-	108	100
110	John	Chen	JCHEN	515.124.4269	28-SEP-87	FI_ACCOUNT	8200	-	108	100

More than 10 rows available. Increase rows selector to view more rows.

12. Display all the records of employees in CSV (comma separated file) form under the heading Employee_ Details

13. There are four coding errors in this statement. Can you identify them?

```
SELECT employee_id, last_name sal x 12 ANNUAL SALARY
```

Ans: - SELECT employee_id, last_name , salary * 12 " Annual Salary" from employees

14. The following SELECT statement executes successfully: (True / False)

```
SELECT last_name, job_id, salary AS Sal FROM employees;
```

Ans: - True.

15. The following SELECT statement executes successfully: (True / False)

```
SELECT * FROM job_grades;
```

Ans: - True;

16. There are four coding errors in this statement. Can you identify them?

```
SELECT employee_id, last_name sal x 12 ANNUAL SALARY FROM employees;
```

Ans: - SELECT employee_id, last_name , salary * 12 " Annual Salary" from employees

17. Show the structure of the DEPARTMENTS table. Select all data from the table.

The screenshot shows a database interface with the following details:

- Autocommit is checked.
- Display dropdown is set to 10.
- SQL input field contains: `desc hr.departments`.
- Run button is visible.
- Results tab is selected.
- Object Type: TABLE Object: DEPARTMENTS.
- Table structure:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPARTMENTS	DEPARTMENT_ID	Number	-	4	0	-	-	-	Primary key column of departments table.
	DEPARTMENT_NAME	Varchar2	30	-	-	-	-	-	A not null column that shows name of a department. Administration, Marketing, Purchasing, Human Resources, Shipping, IT, Executive, Public Relations, Sales, Finance, and Accounting.
	MANAGER_ID	Number	-	6	0	-	✓	-	Manager_Id of a department. Foreign key to employee_Id column of employees table. The manager_Id column of the employee table references this column.
	LOCATION_ID	Number	-	4	0	-	✓	-	Location Id where a department is located. Foreign key to location_Id column of locations table.

Page number: 1 - 4

18. Show the structure of the EMPLOYEES table. Create a query to display the last name, job code, hire date, and employee number for each employee, with employee number appearing first.

Autocommit Display 10

```
select employee_id, last_name, job_id, hire_date
from hr.employees
```

Results Explain Describe Saved SQL History

EMPLOYEE_ID	LAST_NAME	JOB_ID	HIRE_DATE
100	King	AD_PRES	17-JUN-87
101	Kochhar	AD_VP	21-SEP-89
102	De Haan	AD_VP	13-JAN-93
103	Hunold	IT_PROG	03-JAN-90
104	Ernst	IT_PROG	21-MAY-91
105	Austin	IT_PROG	25-JUN-97
106	Pataballa	IT_PROG	05-FEB-98
107	Lorentz	IT_PROG	07-FEB-99
108	Greenberg	FI_MGR	17-AUG-94
109	Faviet	FI_ACCOUNT	16-AUG-94

More than 10 rows available. Increase rows selector to view more rows.

19. Create a query to display unique job codes from the EMPLOYEES table.

Autocommit Display 10

```
select distinct job_id
from hr.employees
```

Results Explain Describe Saved SQL History

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
FI_ACCOUNT
FI_MGR
HR_REP
IT_PROG
MK_MAN

More than 10 rows available. Increase rows selector to view more rows.

20. Create a query to display the last name and salary of employees earning more than 12,000.

Autocommit Display 10 Save Run

```
select last_name,salary
from hr.employees
where salary > 12000
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Russell	14000
Partners	13500
Hartstein	13000

6 rows returned in 0.00 seconds [CSV Export](#)

21. Create a query to display the employee last name and department number for employee number 176.

Autocommit Display 10 Save Run

```
select last_name, department_id
from hr.employees
where employee_id=176
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

LAST_NAME	DEPARTMENT_ID
Taylor	80

1 rows returned in 0.00 seconds [CSV Export](#)

22. Display the employee's last name, job ID, and start date of employees hired between February 20, 1998, and May 1, 1998. Order the query in ascending order by start date

Autocommit Display 10 Save Run

```
select last_name,job_id,hire_date
from hr.employees
where hire_date between '20-FEB-98'and '01-MAY-98'
order by hire_date
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

LAST_NAME	JOB_ID	HIRE_DATE
Fleaur	SH_CLERK	23-FEB-98
Urman	FI_ACCOUNT	07-MAR-98
Matos	ST_CLERK	15-MAR-98
Bloom	SA_REP	23-MAR-98
Taylor	SA_REP	24-MAR-98
Olsen	SA_REP	30-MAR-98
Patel	ST_CLERK	06-APR-98
Livingston	SA_REP	23-APR-98
Walsh	SH_CLERK	24-APR-98

9 rows returned in 0.00 seconds [CSV Export](#)

23. Display the last name and department number of all employees in departments 20 and 50 in alphabetical order by name.

Autocommit Display 10 Save

```
select last_name, department_id
from hr.employees
where department_id=50 or department_id=20
order by last_name
```

Results Explain Describe Saved SQL History

LAST_NAME	DEPARTMENT_ID
Atkinson	50
Bell	50
Bissot	50
Bull	50
Cabrio	50
Chung	50
Davies	50
Dellinger	50
Dilly	50
Everett	50

More than 10 rows available. Increase rows selector to view more rows.

24. Modify Assignment13.sql to list the last 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.

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

Autocommit Display 10 Save

```
select last_name, hire_date
from hr.employees
where hire_date like '%94'
```

Results Explain Describe Saved SQL History

LAST_NAME	HIRE_DATE
Greenberg	17-AUG-94
Faviet	16-AUG-94
Raphaely	07-DEC-94
Mavris	07-JUN-94
Baer	07-JUN-94
Higgins	07-JUN-94
Gietz	07-JUN-94

7 rows returned in 0.02 seconds [CSV Export](#)

26. Display the last name and job title (department_name) of all employees who do not have a manager.

Autocommit Display 10 Save

```
select last_name, d.department_name "Job Title"
from hr.employees e, hr.departments d
where e.manager_id is null and d.department_id=e.department_id
```

Results Explain Describe Saved SQL History

LAST_NAME	Job Title
King	Executive

1 rows returned in 0.00 seconds [CSV Export](#)

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

Autocommit Display 10 Save Run

```
select last_name,salary,commission_pct
from hr.employees
where commission_pct is not null
order by commission_pct desc
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

LAST_NAME	SALARY	COMMISSION_PCT
Russell	14000	.4
McEwen	9000	.35
Sully	9500	.35
King	10000	.35
Doran	7500	.3
Smith	8000	.3
Partners	13500	.3
Abel	11000	.3
Cambrault	11000	.3
Errazuriz	12000	.3

More than 10 rows available. Increase rows selector to view more rows.

28. Display the last names of all employees where the third letter of the first_name is an a.

Autocommit Display 500 Save Run

```
select last_name
from hr.employees
where first_name like '---a%'
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

no data found

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

Autocommit Display 10 Save Run

```
select last_name
from hr.employees
where last_name like '%a%e%' or last_name like '%e%a%'
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

LAST_NAME
Baer
Bates
Colmenares
Davies
De Haan
Faviet
Fleur
Gates
Hartstein
Markle

More than 10 rows available. Increase rows selector to view more rows.

30. Display the last name, job, and salary for all employees whose job is sales representative or stock clerk and whose salary is not equal to 2,500, 3,500, or 7,000.

Autocommit Display 10 Save Run

```
select e.last_name,e.job_id, salary
from hr.employees e,hr.departments d
where e.department_id=d.department_id and e.job_id='ST_CLERK' |and salary not in(2500,3500,700)
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

LAST_NAME	JOB_ID	SALARY
Nayer	ST_CLERK	3200
Mikilineni	ST_CLERK	2700
Landry	ST_CLERK	2400
Markle	ST_CLERK	2200
Bissot	ST_CLERK	3300
Atkinson	ST_CLERK	2800
Olson	ST_CLERK	2100
Mallin	ST_CLERK	3300
Rogers	ST_CLERK	2900
Gee	ST_CLERK	2400

More than 10 rows available. Increase rows selector to view more rows.

31. Write a query to display the current date. Label the column Date.

Autocommit Display 10 Save Run

```
SELECT SYSDATE AS "Date"
FROM DUAL;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Date
27-SEP-22

1 rows returned in 0.00 seconds [CSV Export](#)

35. List the first_name,location id, hire date, job id of all the managers

Autocommit Display 10 Save Run

```
select m.first_name,d.location_id,m.hire_date,m.job_id
from hr.employees e,hr.departments d, hr.employees m
where e.department_id=d.department_id and e.employee_id=m.manager_id
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

FIRST_NAME	LOCATION_ID	HIRE_DATE	JOB_ID
Neena	1700	21-SEP-89	AD_VP
Lex	1700	13-JAN-93	AD_VP
Alexander	1700	03-JAN-90	IT_PROG
Bruce	1400	21-MAY-91	IT_PROG
David	1400	25-JUN-97	IT_PROG
Valli	1400	05-FEB-98	IT_PROG
Diana	1400	07-FEB-99	IT_PROG
Nancy	1700	17-AUG-94	FI_MGR
Daniel	1700	16-AUG-94	FI_ACCOUNT
John	1700	28-SEP-97	FI_ACCOUNT

More than 10 rows available. Increase rows selector to view more rows.

Assignment-4

Q1. Create the following tables:

Client Master Table.

The screenshot shows the Oracle Database Express Edition interface. The title bar says "ORACLE Database Express Edition". The user is "MOHD NASIR". The menu bar has "Home", "Logout", and "Help". The main area is titled "Home > SQL > SQL Commands". A toolbar at the top right has "Save" and "Run" buttons. The SQL editor contains the following code:

```
create table Client Master (client_no varchar2(6), name varchar2(20), city varchar2(15), state varchar2(15),
pincode number(6),bal_due number(10,2))
```

Below the editor, a status bar shows "Results Explain Describe Saved SQL History". The message "Table created." is displayed.

Description Of Client Master Table .

The screenshot shows the Oracle Database Express Edition interface. The title bar says "ORACLE Database Express Edition". The user is "MOHD NASIR". The menu bar has "Home", "Logout", and "Help". The main area is titled "Home > SQL > SQL Commands". A toolbar at the top right has "Save" and "Run" buttons. The SQL editor contains the following code:

```
desc client master
```

Below the editor, a status bar shows "Results Explain Describe Saved SQL History". A table titled "Object Type TABLE Object CLIENT_MASTER" is displayed, showing the columns and their properties:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CLIENT_MASTER	CLIENT_NO	Varchar2	6	-	-	-	✓	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	CITY	Varchar2	15	-	-	-	✓	-	-
	STATE	Varchar2	15	-	-	-	✓	-	-
	PINCODE	Number	-	6	0	-	✓	-	-
	BAL_DUE	Number	-	10	2	-	✓	-	-

At the bottom right of the table, it says "1 - 6".

Q2 (i).Data For Client Table.

The screenshot shows the Oracle Database Express Edition interface. The title bar says "ORACLE Database Express Edition". The user is "MOHD NASIR". The menu bar has "Home", "Logout", and "Help". The main area is titled "Home > SQL > SQL Commands". A toolbar at the top right has "Save" and "Run" buttons. The SQL editor contains the following code:

```
insert all
into client_master values ('0001','Ivan', 'Bombay', 'Maharashtra',400054, 15000)
into client_master values ('0002','Vandana', 'Madras', 'Tamilnadu',780001, 0)
into client_master values ('0003','Pramada', 'Bombay', 'Maharashtra', 400057, 5000)
into client_master values ('0004','Basu', 'Bombay', 'Maharashtra', 400056,0)
into client_master values ('0005','Ravi', 'Bombay', null , 100001, 0)
into client_master values ('0006','Amit', 'Bombay', 'Maharashtra',400050, 15000)
select * from dual
```

Below the editor, a status bar shows "Results Explain Describe Saved SQL History". The message "6 row(s) inserted." is displayed.

Product Master.

Autocommit Display 10

```
create table Product_Master(product_no varchar2(10), Description varchar2(10), profit_percent number(10),
unit_measure varchar2(5), qty_on_hand number(5), reorder_lvl number(5), sell_price number (6,2))
```

Results Explain Describe Saved SQL History

Table created.

Description of Product Master Table.

Autocommit Display 10

```
describe Product_Master
```

Results Explain Describe Saved SQL History

Object Type TABLE Object PRODUCT_MASTER

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT_MASTER	PRODUCT_NO	Varchar2	10	-	-	-	✓	-	-
	DESCRIPTION	Varchar2	10	-	-	-	✓	-	-
	PROFIT_PERCENT	Number	-	10	0	-	✓	-	-
	UNIT_MEASURE	Varchar2	5	-	-	-	✓	-	-
	QTY_ON_HAND	Number	-	5	0	-	✓	-	-
	REORDER_LVL	Number	-	5	0	-	✓	-	-
	SELL_PRICE	Number	-	6	2	-	✓	-	-

1 - 7

Q2. (ii) Data For Product Master Table.

Autocommit Display 10

```
insert all
into product_master values ('P00001','floppies', 5, 'piece',100, 20,525)
into product_master values ('P03453','Monitors', 6, 'piece',10,3,9000)
into product_master values ('P06734','Mouse', 5, 'piece',20, 5,1050)
into product_master values ('P07865','floppies', 6, 'piece',100, 20,525)
into product_master values ('P07868','Keyboards', 2, 'piece',10, 3,3150)
into product_master values ('P07885','CD Drive', 2.5, 'piece',10, 2,5250)
into product_master values ('P07965','540 HDD', 4, 'piece',10, 3,8400)
into product_master values ('P07975','1.44 Drive', 3, 'piece',10, 3,1050)
into product_master values ('P08865','1.22 Drive', 5, 'piece',2, 3,1050)
select * from dual
```

Results Explain Describe Saved SQL History

9 row(s) inserted.

Q3: - On the basis of above two tables answer the following Questionnaires:

- (i) Find out the names of all the clients.

Autocommit Display 10

```
select *
from client_master
where client_no in(0001,0002)
```

Results Explain Describe Saved SQL History

CLIENT_NO	NAME	CITY	STATE	PINCODE	BAL_DUE
0001	Ivan	Bombay	Maharashtra	400054	15000
0002	Vandana	Madras	Tamilnadu	780001	0

2 rows returned in 0.00 seconds [CSV Export](#)

(ii). Retrieve the list of names and cities of all the clients.

```
Autocommit Display | 10
select name, city
from client_master
```

Save Run

Results Explain Describe Saved SQL History

NAME	CITY
Ivan	Bombay
Vandana	Madras
Pramada	Bombay
Basu	Bombay
Ravi	Bombay
Amit	Bombay

6 rows returned in 0.00 seconds CSV Export

(ii) Display the information for client no 0001 and 0002.

```
Autocommit Display | 10
select *
from client_master
where client_no in(0001,0002)
```

Save Run

Results Explain Describe Saved SQL History

CLIENT_NO	NAME	CITY	STATE	PINCODE	BAL_DUE
0001	Ivan	Bombay	Maharashtra	400054	15000
0002	Vandana	Madras	Tamilnadu	780001	0

2 rows returned in 0.00 seconds CSV Export

(iii) Find all the products whose sell price is greater than 5000.

```
Autocommit Display | 10
select *
from product_master
where sell_price>5000
```

Save Run

Results Explain Describe Saved SQL History

PRODUCT_NO	DESCRIPTION	PROFIT_PERCENT	UNIT_MEASURE	QTY_ON_HAND	REORDER_LVL	SELL_PRICE
P03453	Monitors	6	piece	10	3	9000
P07885	CD Drive	3	piece	10	2	5250
P07965	540 HDD	4	piece	10	3	8400

3 rows returned in 0.00 seconds CSV Export

ASSIGNMENT 5

Ques1. Create the following tables.

Creation Of Sales_Master Table.

The screenshot shows the Oracle Database Express Edition interface. The title bar says "ORACLE® Database Express Edition". The user is "MOHD_NASIR". The menu bar includes "Home", "Logout", and "Help". The main area shows the SQL Commands page with the following SQL code:

```
create table sales_master (salesman_no varchar2(6), constraint sm_pk primary key(salesman_no), sal_name varchar2(20) NOT NULL, Address varchar2(50) NOT NULL, city varchar2(20), state varchar2(20), pincode number(6), sal_amt number(8,2) NOT NULL check (sal_amt>0), tgt_to_get number(6,2) NOT NULL check (tgt_to_get>0), ytd_sales number (6,2) NOT NULL check (ytd_sales>0), remarks varchar2(30))
```

Below the code, the status message "Table created." is displayed.

Insert Data in Sales_Master Table.

The screenshot shows the Oracle Database Express Edition interface. The title bar says "ORACLE® Database Express Edition". The main area shows the SQL Commands page with the following SQL code:

```
insert all
into sales_master values('500001','Kiran','A/12 Worli','Bombay','Maharashtra',400002,3000,100,50,'Good')
into sales_master values('500002','Manish','65, Narima','Bombay','Maharashtra',400001,3000,200,100,'Good')
into sales_master values('500003','Ravi','P-7 Bandra','Bombay','Maharashtra',400032,3000,200,100,'Good')
into sales_master values('500004','Aashish','A/5 Juhu','Bombay','Maharashtra',400044,3000,200,100,'Good')
select * from dual
```

Below the code, the status message "4 row(s) inserted." is displayed.

Creation Of Sales_Order.

The screenshot shows the Oracle Database Express Edition interface. The title bar says "ORACLE® Database Express Edition". The main area shows the SQL Commands page with the following SQL code:

```
insert all
into sales_order values('019001','12-jan-96','0001','DDN','500001','F','N','20-jan-96','Ip')
into sales_order values('019002','25-jan-96','0002','DDN','500002','P','N','27-jan-96','C')
into sales_order values('016865','18-feb-96','0003','DDN','500003','F','Y','20-feb-96','F')
into sales_order values('019003','03-apr-96','0001','DDN','500001','F','Y','07-apr-96','F')
into sales_order values('046866','20-may-96','0004','DDN','500002','P','N','22-may-96','C')
into sales_order values('010008','24-may-96','0005','DDN','500004','F','N','26-may-96','Ip')
select * from dual
```

Below the code, the status message "6 row(s) inserted." is displayed.

Insert Data in Sales_Order Table.

The screenshot shows the Oracle Database Express Edition interface. The title bar says "ORACLE® Database Express Edition". The main area shows the SQL Commands page with the following SQL code:

```
create table sales_order (s_order_no varchar2(6), constraint so_pk primary key(s_order_no), s_order_date date, client_no varchar2(25), foreign key (client_no) references client_master(client_no), Dely_add varchar2(6), salesman_no varchar2(6), foreign key (salesman_no) references sales_master(salesman_no), dely_type char(1) default 'f', billed_y_n char(1), dely_date date, order_status varchar2(10))
```

Below the code, the status message "Table created." is displayed.

Creation Of Sales_Order_Table.

```
Autocommit Display 10 Save Run
create table sales_order_details (s_order_no varchar (6), foreign key (s_order_no) references sales_order(s_order_no),product_no varchar (10), foreign key (product_no) references product_master(product_no), primary key(s_order_no,product_no),qty_order number(8), qty_disp number(8),product_rate number(8,2))

Results Explain Describe Saved SQL History
```

Table created.

Insert Data in Sales_Order_Details.

```
Autocommit Display 10 Save Run
insert all
into sales_order_details values('019001','P00001',4,4,525)
into sales_order_details values('019001','P07965',2,1,8400)
into sales_order_details values('019001','P07885',2,1,5250)
into sales_order_details values('019002','P00001',10,0,525)
into sales_order_details values('046866','P07868',3,3,3150)
into sales_order_details values('046866','P07885',10,10,5250)
into sales_order_details values('019003','P00001',4,4,1050)
into sales_order_details values('019003','P03453',2,2,1050)
into sales_order_details values('046866','P06734',1,1,12000)
into sales_order_details values('046866','P07965',1,0,8400)
into sales_order_details values('010008','P07975',1,0,100)
into sales_order_details values('010008','P00001',10,5,525)
select * from dual

Results Explain Describe Saved SQL History
```

12 row(s) inserted.

ASSIGNMENT 6

Ques1. Create the following tables.

Creation Of Challan_Header Table.

```
Autocommit Display 10 Save Run
create table challan_header(challan_no varchar2(6), constraint ch_pk primary key(challan_no), s_order_no
varchar2(6), foreign key (s_order_no) references sales_order(s_order_no), challan_date date not null, build_yn
char(1) default 'N')

Results Explain Describe Saved SQL History
Table created.
```

Creation Of Challan Details Table.

```
Autocommit Display 10 Save Run
create table challan_details(challan_no varchar2(6),product_no varchar2(10), foreign key (challan_no)references
challan_header(challan_no),foreign key (product_no) references product_master(product_no), primary
key(product_no,challan_no), qty_disp number(4))

Results Explain Describe Saved SQL History
Table created.
```

Ques2. Insert the following values into the challan_heard and challan_details table.

Insert data Challan_Header Table.

```
Autocommit Display 10 Save Run
insert all
into challan_header values('CH9001','019001','12-DEC-95','Y')
into challan_header values('CH865','046866','12-NOV-95','Y')
into challan_header values('CH3965','010008','12-OCT-95','Y')
select * from dual

Results Explain Describe Saved SQL History
3 row(s) inserted.
```

Insert Data in Challan_Details Table.

```
Autocommit Display 10 Save Run
insert all
into challan_details values ('CH9001','P00001',4)
into challan_details values ('CH9001','P07965',1)
into challan_details values ('CH9001','P07885',1)
into challan_details values ('CH865','P07868',3)
into challan_details values ('CH865','P03453',4)
into challan_details values ('CH865','P00001',10)
into challan_details values ('CH3965','P00001',5)
into challan_details values ('CH3965','P07975',2)
select * from dual

Results Explain Describe Saved SQL History
8 row(s) inserted.
```

Ques3. Using the table Assignment 4 & 5 answer te following Questionnaires.

(i). Select Product_No, Total Qty-Ordered for each product.

Autocommit Display 10 Save Run

```
select product_no,sum(qty_order) as "Total Qty_Ordered"
from sales_order_details
group by product_no
order by product_no
```

Results Explain Describe Saved SQL History

PRODUCT_NO	Total Qty_Ordered
P00001	28
P03453	2
P06734	1
P07868	3
P07885	12
P07965	3
P07975	1

(ii). Select Product_No

Autocommit Display 15 Save Run

```
select d.product_no , d.qty_order,m.description
from sales_order_details d, product_master m
where d.product_no=m.product_no
```

Results Explain Describe Saved SQL History

PRODUCT_NO	QTY_ORDER	DESCRIPTION
P00001	4	floppies
P07965	2	540 HDD
P07885	2	CD Drive
P00001	10	floppies
P07868	3	Keyboards
P07885	10	CD Drive
P00001	4	floppies
P03453	2	Monitors
P06734	1	Mouse
P07965	1	540 HDD
P07975	1	1.44 Drive
P00001	10	floppies

12 rows returned in 0.00 seconds CSV Export

Assignment 8

Consider the following relations:

Faculty (fid: integer, fname: string, deptid: integer)

User: MOHD NASIR

Home > SQL > SQL Commands

Autocommit Display 10

```
create table faculty(fid varchar(6), constraint fi_pk primary key(fid), fname varchar2(20))
```

Results Explain Describe Saved SQL History

Table created.

Student (snum: integer, sname: string, major: string, level: string, age: integer, fid: integer)

User: MOHD NASIR

Home > SQL > SQL Commands

Autocommit Display 10

```
create table student(snum varchar2(20),constraint snum_pk primary key (snum), sname varchar2(20), major varchar2(20), slevel varchar2(10), age number(2) check (age>0), fid varchar2(6),foreign key (fid) references faculty(fid))
```

Results Explain Describe Saved SQL History

Table created.

Class (name: string, meets at: string, room: string, fid: integer)

User: MOHD NASIR

Home > SQL > SQL Commands

Autocommit Display 10

```
create table class(cname varchar2(20),constraint cn_pk primary key (cname), meets varchar2(20), room varchar2(20), fid varchar2(6),foreign key (fid) references faculty(fid))
```

Results Explain Describe Saved SQL History

Table created.

Enrolled (snum: integer, cname: string)

User: MOHD NASIR

Home > SQL > SQL Commands

Autocommit Display 10

```
create table enrolled(fid varchar2(6),foreign key (fid) references faculty(fid), cname varchar2(20),foreign key (cname) references class(cname),constraint en_pk primary key (cname,fid))
```

Results Explain Describe Saved SQL History

Table created.

Autocommit Display 10 Save Run

```
desc faculty
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Object Type TABLE Object FACULTY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
FACULTY	FID	Varchar2	6	-	-	1	-	-	-
	FNAME	Varchar2	20	-	-	-	✓	-	-

1 - 2

Description Of Student Table.

Autocommit Display 10 Save Run

```
desc student
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Object Type TABLE Object STUDENT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STUDENT	SNUM	Varchar2	20	-	-	1	-	-	-
	SNAME	Varchar2	20	-	-	-	✓	-	-
	MAJOR	Varchar2	20	-	-	-	✓	-	-
	SLEVEL	Varchar2	10	-	-	-	✓	-	-
	AGE	Number	-	2	0	-	✓	-	-
	FID	Varchar2	6	-	-	-	✓	-	-

1 - 6

Description Of Class Table.

Autocommit Display 10 Save Run

```
desc class
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Object Type TABLE Object CLASS

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CLASS	CNAME	Varchar2	20	-	-	1	-	-	-
	MEETS	Varchar2	20	-	-	-	✓	-	-
	ROOM	Varchar2	20	-	-	-	✓	-	-
	FID	Varchar2	6	-	-	-	✓	-	-

1 - 4

Description Of Enrolled Table.

Autocommit Display 10 Save Run

```
desc enrolled
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Object Type TABLE Object ENROLLED

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ENROLLED	FID	Varchar2	6	-	-	2	-	-	-
	CNAME	Varchar2	20	-	-	1	-	-	-

1 - 2

Autocommit Display 10

```
insert all
into student values('20021595', 'Rajkumar', 'CSE','JR',19,'F0001')
into student values('20021596', 'Raju Rastogi', 'CSE','UG',21,'F0003')
into student values('20021597', 'Faisal Saifi', 'CST','UG',22,'F0004')
into student values('20021598', 'Mohd Anas', 'CE','JR',19,'F0001')
into student values('20021599', 'Neelam', 'CSE','Ug',23,'F0001')
select * from dual
```

Results Explain Describe Saved SQL History

5 row(s) inserted.

Autocommit Display 10

```
insert all
into faculty values('F0001','Harsith')
into faculty values('F0002', 'Mohd Nasir')
into faculty values('F0003', 'Deepak Kumar')
into faculty values('F0004', 'Rashi Singh')
into faculty values('F0005', 'Vaibhav Pratap')
select * from dual
```

Results Explain Describe Saved SQL History

5 row(s) inserted.

Autocommit Display 10

```
insert all
into class values ('DBMS','9:00-9:55', 'R128', 'F0001')
into class values ('System Software','9:55-10:50', 'R128', 'F0002')
into class values ('Operating System','9:00-9:55', 'R127', 'F0003')
into class values ('CBNST','9:00-9:55', 'R126', 'F0004')
into class values ('JAVA','9:00-9:55', 'R127', 'F0005')
select* from dual]
```

Results Explain Describe Saved SQL History

5 row(s) inserted.