```
alter user hr identified by hr account unlock;
/
SELECT l.first_name,
  l.last_name,
  d.department_name,
  e.city,
  e.street_address
FROM employees l
JOIN departments d
ON (l.department_id= d.department_id)
JOIN locations e
ON (e.location_id = d.location_id);
SELECT * FROM locations;

/

SELECT l.first_name,
  l.last_name,
  d.department_name,
  e.city
FROM employees l
JOIN departments d
ON (l.department_id= d.department_id)
JOIN locations e
ON (e.location_id     = d.location_id)
WHERE d.department_id = 100;

/

SELECT l.first_name,
  l.last_name,
  d.department_name,
  e.city
FROM employees l
JOIN departments d
ON (l.department_id= d.department_id)
JOIN locations e
ON (e.location_id     = d.location_id)
WHERE d.department_id = 100;
SELECT * FROM jobs;
SELECT a.first_name,
  a.last_name,
  b.job_title,
  a.salary,
  b.min_salary,
  b.max_salary
FROM employees a
JOIN jobs b
ON a.salary BETWEEN b.min_salary AND b.max_salary;

/

SELECT worker.first_name
  || ' '
```

```
  ||worker.last_name "Employee Name",
  manager.first_name
  || ' '
  || manager.last_name "Manager Name",
  worker.employee_id,
  worker.manager_id,
  manager.employee_id "Manager Employee Id"
FROM employees worker
JOIN employees manager
ON (worker.manager_id = manager.employee_id)
ORDER BY worker.employee_id;

/

--Left outer join
SELECT e.first_name,
  e.last_name,
  e.department_id,
  d.department_id,
  d.department_name
FROM employees e
LEFT OUTER JOIN departments d
ON (e.department_id = d.department_id) ;

/
--Right Outer Join
SELECT e.first_name,
  e.last_name,
  e.department_id,
  d.department_id,
  d.department_name
FROM employees e
RIGHT OUTER JOIN departments d
ON (e.department_id = d.department_id);

/
--Full Outer Join
SELECT e.first_name,
  e.last_name,
  e.department_id,
  d.department_id,
  d.department_name
FROM employees e
FULL OUTER JOIN departments d
ON (e.department_id = d.department_id);

/
--Cross Join
SELECT e.first_name,
  e.last_name,
  e.department_id,
  d.department_id,
  d.department_name
FROM employees e
```

```
CROSS JOIN departments d;

/

SELECT first_name,
  last_name,
  salary
FROM employees
WHERE salary >
  (SELECT salary
  FROM employees
  WHERE first_name = 'Michael'
  AND last_name    = 'Hartstein'
  );

/

SELECT first_name,
  last_name,
  department_id
FROM employees
WHERE department_id =
  (SELECT department_id
  FROM employees
  WHERE first_name = 'Michael'
  AND last_name    = 'Hartstein'
  );

/

SELECT first_name,
  last_name,
  salary
FROM employees
WHERE salary <
  (SELECT salary
  FROM employees
  WHERE first_name = 'Michael'
  AND last_name    = 'Hartstein'
  )
AND department_id =
  (SELECT department_id
  FROM employees
  WHERE first_name = 'Michael'
  AND last_name    = 'Hartstein'
  );

/

SELECT first_name,
  last_name,
  hire_date
FROM employees
WHERE hire_date=
```

```
  (SELECT MIN(hire_date) FROM employees
  );

/

SELECT first_name,
  last_name,
  hire_date
FROM employees
WHERE hire_date=
  (SELECT MIN(hire_date) FROM employees GROUP BY department_id
  );

/
--IN Operator
SELECT first_name,
  last_name,
  department_id,
  salary
FROM employees
WHERE salary IN
  (SELECT MIN(salary) FROM employees GROUP BY department_id
  );

/
--ANY
SELECT first_name,
  last_name,
  job_id,
  salary
FROM employees
WHERE salary > ANY
  (SELECT salary FROM employees WHERE job_id = 'SA_MAN'
  );
SELECT MIN(salary) FROM employees WHERE job_id = 'SA_MAN' /
SELECT first_name,last_name,department_id, salary FROM employees;

/

SELECT first_name,
  last_name,
  job_id,
  salary
FROM employees
WHERE salary > ALL
  (SELECT salary FROM employees WHERE job_id = 'SA_MAN'
  );
/
SELECT first_name,
  last_name,
  job_id,
  salary
FROM employees
WHERE salary > ALL
```

```
  (SELECT salary FROM employees WHERE job_id = 'SA_MAN'
  );

/

SELECT first_name,
  last_name,
  job_id,
  salary
FROM employees
WHERE (salary, job_id) IN
  (SELECT MAX(salary), job_id FROM employees GROUP BY job_id
  )
ORDER BY salary DESC;
/
SELECT job_id FROM job_history
INTERSECT
SELECT job_id FROM employees WHERE department_id = 80 ;-- where manager_id
= 100;
/
SELECT * FROM JOB_HISTORY;
/
SELECT job_id,
  department_id,
  first_name
FROM employees
WHERE department_id = 80
UNION
SELECT job_id,department_id,NULL FROM job_history ORDER BY department_id;
/
SELECT * FROM JOB_HISTORY;
INSERT
INTO job_history
  (
    employee_id,
    start_date,
    end_date,
    job_id,
    department_id
  )
  VALUES
  (
    120,
    to_date('01/05/03','DD/MM/YY'),
    sysdate,
    'IT_PROG',
    60
  );
INSERT INTO jobs VALUES
  ('PR_MGR', 'Project Manager',11000, 18000
  );
SELECT * FROM DEPARTMENTS;
INSERT
INTO DEPARTMENTS
```

```
  (
    department_id,
    department_name
  )
  VALUES
  (
    290,
    'Transportation'
  );
CREATE TABLE IT_PROGRAMMERS AS
SELECT * FROM employees WHERE 1=2;
INSERT INTO IT_PROGRAMMERS
SELECT * FROM employees WHERE job_id LIKE 'IT_PROG';
CREATE TABLE EMPLOYEES_COPY AS
SELECT * FROM employees WHERE 1=2;
INSERT INTO IT_PROGRAMMERS
SELECT * FROM employees WHERE job_id LIKE 'IT_PROG';
SELECT * FROM EMPLOYEE_ADDRESSES;

/

CREATE TABLE EMPLOYEE_ADDRESSES(employee_id NUMBER, FIRST_NAME
VARCHAR2(50), LAST_NAME VARCHAR2(50), ADDRESS VARCHAR2(200));
INSERT INTO EMPLOYEE_ADDRESSES
SELECT employee_id,first_name,
       last_name,city || ' '  || street_address address
    FROM EMPLOYEES
    JOIN departments USING (department_id)
    JOIN locations USING (location_id);

/
UPDATE employees
SET salary = 50000;

/

UPDATE employees_copy
SET salary = 50000;

/
UPDATE employees_copy
SET (salary,commission_pct) = (SELECT MAX(salary),MAX(commission_pct) FROM
employees)
WHERE job_id = 'IT_PROG';

insert into employees_copy select * from employees;
--Savepoint Statement
select * from employees_copy;
delete from employees_copy where employee_id = '100';
savepoint a;

delete from employees_copy where employee_id = '101';
savepoint b;
```

```
delete from employees_copy where employee_id = '102';
savepoint c;

delete from employees_copy where employee_id = '103';
savepoint d;
delete from employees_copy where employee_id = '104';
select * from employees_copy;
rollback;

/
--- For Update Statement ---
SELECT * FROM EMPLOYEES WHERE JOB_ID = 'IT_PROG' FOR UPDATE NOWAIT;
select * from departments where department_id = 60;
/
select employee_id,first_name, last_name,email,department_id from employees
e join departments d using (department_id) where location_id = 1400 ;

select first_name, last_name, salary from employees e join departments d
using (department_id) where location_id = 1400 for update of
e.salary,d.location_id;

/

CREATE TABLE my_employees
(employee_id NUMBER(3),
 first_name VARCHAR2(50),
 last_name VARCHAR2(50),
 hire_date DATE DEFAULT sysdate);

/

CREATE TABLE employees_copy2(first_name, last_name, salary)
AS SELECT first_name, last_name, salary FROM employees;

/
CREATE TABLE employees_copy2(name, surname, annual_salary)
AS SELECT first_name, last_name, 12*salary FROM employees;

/

SELECT * FROM EMPLOYEES_COPY;

/
DESC EMPLOYEES_COPY;

/

CREATE TABLE employees_copy2(name, surname)
AS SELECT first_name, last_name, 12*salary FROM employees;

/
SELECT * FROM EMPLOYEES_COPY;
ALTER TABLE employees_copy
MODIFY commission_pct DEFAULT 0;
```

```
/
ALTER TABLE employees_copy
DROP   (fax_number,
        password);
/

ALTER TABLE employees_copy
DROP column fathers_name;

/

ALTER TABLE employees_copy
ADD  (fax_number   NUMBER,
      fathers_name VARCHAR2(50),
      password     VARCHAR2(10) DEFAULT 'abc123');
/

TRUNCATE TABLE EMPLOYEES_COPY;

/
INSERT INTO EMPLOYEES_COPY SELECT * FROM EMPLOYEES;

/

FLASHBACK TABLE HR.EMPLOYEES_COPY TO BEFORE TRUNCATE;

/

ALTER TABLE EMPLOYEES_COPY
ENABLE ROW MOVEMENT;

/

FLASHBACK TABLE EMPLOYEES_COPY
TO TIMESTAMP (SYSTIMESTAMP - INTERVAL '3' minute);

/

select * from EMPLOYEES_COPY;
ALTER TABLE EMPLOYEES_COPY RENAME TO EMPLOYEES_COPY_NEW;
ALTER TABLE EMPLOYEES_COPY_NEW RENAME TO EMPLOYEES_COPY;
/
RENAME COLUMN EMPLOYEES_COPY.FATHERS_NAME to EMPLOYEES_COPY.FATHER;
/
SELECT *  FROM HR.EMPLOYEES_COPY;
alter table
EMPLOYEES_COPY.FAX_NUMER
rename to
   new_table_name;
/
  ALTER TABLE EMPLOYEES_COPY RENAME COLUMN PASSWORD TO PASS;
/
  SELECT * FROM EMPLOYEES_COPY;
```

```
  RENAME EMPLOYEES_COPY TO RETIRED_EMPLOYEES;
  ALTER TABLE RETIRED_EMPLOYEES RENAME TO EMPLOYEES_COPY;
/
  COMMENT ON table employees_copy IS 'Employees';
/

CREATE TABLE MANAGERS
( employee_id NUMBER CONSTRAINT constraint_name CONSTRAINT_TYPE,
  first_name VARCHAR2(50),
  last_name VARCHAR2(50),
  department_id NUMBER,
  [CONSTRAINT constraint_name] CONSTRAINT_TYPE (column_name1, ...)
);
/

select * from user_col_comments;

/
DROP TABLE MANAGERS;
CREATE TABLE MANAGERS2
(employee_id   NUMBER,
  first_name    VARCHAR2(50) NOT NULL,
  last_name     VARCHAR2(50) CONSTRAINT lname_cons NOT NULL,
  department_id NUMBER NOT NULL,
  phone_number  VARCHAR2(11) UNIQUE,
  CONSTRAINT fname_cons NOT NULL(employee_id)
);

  CREATE TABLE MANAGERS
( employee_id   NUMBER NOT NULL,
  first_name    VARCHAR2(50),
  last_name     VARCHAR2(50) CONSTRAINT lname_cons NOT NULL,
  department_id NUMBER NOT NULL
);

  ALTER TABLE MANAGERS ADD CONSTRAINT PK_MANAGERS PRIMARY KEY(employee_id)
;

  CREATE TABLE MANAGERS
( employee_id NUMBER,
  first_name  VARCHAR2(50),
  last_name   VARCHAR2(50),
  CONSTRAINT PK_MANAGERS PRIMARY KEY );


  CREATE TABLE MANAGERS
( employee_id   NUMBER,
  first_name    VARCHAR2(50) UNIQUE,
  last_name     VARCHAR2(50) CONSTRAINT lname_cons UNIQUE,
  department_id NUMBER NOT NULL,
  phone_number  VARCHAR2(11) UNIQUE NOT NULL,
  CONSTRAINT department_cons UNIQUE(department_id),
  CONSTRAINT COMPOSED_UNIQUE UNIQUE (employee_id, first_name,last_name)
);
```

```
CREATE TABLE MANAGERS2
( employee_id   NUMBER CONSTRAINT PK_MANAGERS PRIMARY KEY,
  first_name    VARCHAR2(50),
  last_name     VARCHAR2(50)
);
CREATE TABLE MANAGERS
( employee_id NUMBER,
  first_name  VARCHAR2(50),
  last_name   VARCHAR2(50),
  CONSTRAINT PK_MANAGERS PRIMARY KEY(EMPLOYEE_ID) );

/

CREATE TABLE MANAGERS
(manager_id   NUMBER CONSTRAINT PK_MANAGERS PRIMARY KEY,
 first_name    VARCHAR2(50),
 last_name     VARCHAR2(50),
 department_id NUMBER,
 CONSTRAINT emp_man_fk FOREIGN KEY (manager_id) REFERENCES employees
(employee_id)
 ON DELETE SET NULL
);

DROP TABLE managers;

/

CREATE TABLE MANAGERS
(manager_id    NUMBER CONSTRAINT PK_MANAGERS PRIMARY KEY,
 first_name    VARCHAR2(50),
 last_name     VARCHAR2(50),
 department_id NUMBER,
 salary          NUMBER,
 CONSTRAINT min_salary CHECK( salary>0), CHECK (department_id > 0)
);

/

CREATE TABLE MANAGERS
(manager_id    NUMBER CONSTRAINT PK_MANAGERS PRIMARY KEY,
 first_name    VARCHAR2(50),
 last_name     VARCHAR2(50),
 department_id NUMBER,
 salary          NUMBER,
 CHECK (department_id >),
 CONSTRAINT min_salary CHECK( salary>0 )
);
```