



K.S. GROUP OF INSTITUTIONS
K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

#15, Mallasandra, Near Vajarahalli, Off. Kanakapura Road, Bengaluru - 560 109

PRACTICAL RECORD BOOK

NAME : HRISHIKESH.B.S

Class : IV CSE (A)

Lab : Database Management Systems

USN :

1	K	G	2	2	C	S	0	5	0
---	---	---	---	---	---	---	---	---	---



K.S. GROUP OF INSTITUTIONS
K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

#15, Mallasandra, Near Vajarahalli, Off. Kanakapura Road, Bengaluru - 560 109



KSSEM
K.S. SCHOOL OF ENGINEERING AND MANAGEMENT

Laboratory Certificate

This is to certify that

Mr. / Ms. ...**HRISHIKESH.BS**.....
has satisfactorily completed the course of experiment in
..Database..Management..Systems..... Laboratory, Code ..**BCS403**..
Prescribed by Visvesvaraya Technological University, Belagavi
for the**IV**..... Semester B.E.**CSE**..... branch in this
college during in the academic year 20 ..**23**.... - 20 ..**24**....

Name of the Candidate :**HRISHIKESH B S**.....

USN :**IKG22CS050**..... Lab (with code)**BCS403**.....

Marks	
Maximum	Obtained
25	23

Date :

21/8/24
Signature of the
Teacher

Head of the Department

HOD

Department of Computer Science Engineering
K.S School of Engineering & Management
Bangalore-560109

Particulars of the Experiments Performed

CONTENTS

[illegible]

DATE 16/05/2024

EXPT TITLE: Employee database

EXP. NO. 1

PAGE NO.

1

Create a table called Employee & execute the following:

Employee (EMPNO, ENAME, JOB, MANAGER-NO, SAL, COMMISSION)

1. Create a user & grant all permissions to the user.
2. Insert any 3 records in the employee table contains attributes EMPNO, ENAME, JOB, MANAGER-NO, SAL, COMMISSION & use rollback check the result
3. Add primary key constraint & not null constraint to the employee table
4. Insert null values to the employee table & verify the result.

1. Create a user & grant all permission to the user

Conn/as Sysdba

Create user

grant dba to CS0540

Conn,

Enter username : CS050

Enter password : 654321

2. Insert any 3 records in the employee table contains attributes EMPNO, ENAME, JOB, MANAGER-NO, SAL, COMMISSION & use rollback - check the results

Create table employee (EMPNO. number (4), ename Varchar(20), job Varchar(20), manager-no number (4), sal number (8,2), commission number (6,2));

desc employee

Create d

All permission granted
Connected

Table created

Table	column	Datatype	Precision	Scale	Primary key	Nullable	Default	Comments
Employee	EMPNO	-	4	0	-	✓	-	-
	ENAME	20	-	-	-	✓	-	-
	JOB	20	-	-	-	✓	-	-
	MANAGER	-	4	0	-	✓	-	-
	SAL	-	8	2	-	✓	-	-
	COMMISSION	-	6	2	-	✓	-	-

Insert into employee values (1, 'Stonebridge', 'Data-Analyst', 100, 100000, 3000);

Insert into employee values (2, 'Scott', 'web developer', 200, 70000, 3000);

Insert into employee values (3, 'Crawford', 'Data Scientist', 300, 75000, 3000);

Select * from employee;

Roll back;

Select * from employee;

3. Add ~~primary~~ key constraint & not null constraint to the employee table;
alter table employee add constraints, primary key (EMPNO);

alter table employee add constraint not null (SAL);

desc employee;

1 How inserted

1 How inserted

1 How inserted

EMPNO	ENAME	JOB	MANAGER-NO	SAL	COMMISSION
1	Stonebridge	Data-Analyst	100	100000	3000
2	Scott	web developer	200	70000	3000
3	Crawford	Data Scientist	300	75000	3000

How affected

EMPNO	ENAME	JOB	MANAGER-NO	SAL	COMMISSION
1	Stonebridge	Data-Analyst	100	100000	3000
2	Scott	web-developer	200	70000	3000

2 rows affected

2 rows affected

Table	column	Datatype	length	precision	Scale	Primary key	Nullable	default	Comm
Employee	EMPNO	Number	-	4	0	Key	-	-	-
	ENAME	Varchan	20	-	-	-	✓	-	-
	JOB	Varchan	20	-	-	-	✓	-	-
	MGIR-NO	Number	-	4	0	-	✓	-	-
	SAL	Number	-	8	2	-	✓	-	-
	COMMISSION	Number	-	6	2	-	✓	-	-

Insert Null values to the employee table & verify the result

Update employee set commission = Null where empno = 2;

select * from employee;

25/12

10

How updated

EMPNO	ENAME	JOB	MANAGER.NO	SAL	COMMISSION
1	Stonebridge	Data Analyst	100	100000	3000
2	Scott	web developer	200	70000	-

Create a table Employee that contains attribute EMPNO, ENAME, JOB, MGR, SAL & execute the following

1. Add a column commission with domain to the employee table.
2. Insert 5 records into the table.
3. Update the column details of job.
4. Rename the column of employee table using alter command
5. Delete the employee where EMPNO & 105

Create table employee (EMPNO number(4), ENAME varchar(20), JOB varchar(50), MGR number(4), SAL number(8,2));

desc employees;

1. Add column commission with domain to the employee table

Create Domain Domain-Constraint number(6,2);

alter table employee add commission Domain- commission;

desc employee;

Table created

Table	column	Datatype	length	Precision	Scale	Primary key	Nullable	default	Commission
Emplo yee	EMPNO	Number	-	4	0	-	✓	-	-
	ENAME	Varchar	20	-	-	-	✓	-	-
	JOB	Varchar	20	-	-	-	✓	-	-
	MGR	Number	-	4	0	-	✓	-	-
	SAL	Number	-	8	2	-	✓	-	-

1-6

Table	column	Datatype	length	Precision	Scale	Primary key	Nullable	default	Commission
Empl- oyee	EMPNO	Number	-	4	0	-	✓	-	-
	ENAME	Varchar	20	-	-	-	✓	-	-
	JOB	Varchar	50	-	-	-	✓	-	-
	MGR	Number	-	4	0	-	✓	-	-
	SAL	Number	-	8	2	-	✓	-	-
	COMMISSION	Number	-	6	2	-	✓	-	-

DATE 27/05/2024

EXP. NO. 2

EXPT TITLE: Queries using DDL & DML Commands

PAGE NO. 5

2. Insert 5 records into the table

insert into employee values (101, 'Stonebridge', 'Data Analyst', 10, 100000.60,
5000.90);

insert into employee values (102, 'Scott', 'Software Developer', 100, 70000.96,
3000.76);

insert into employee values (103, 'Grawford', 'Analyst', 90, 50000.00,
3000.54);

insert into employee values (104, 'John', 'Webdeveloper', 70, 90000.00
5000.64);

insert into employee values (105, 'Rowinson', 'HR', 80, 200000.00, 7000.91);

~~select * from employee;~~

1 row inserted

1 row inserted

1 row inserted

1 row inserted

1 row inserted

EMPNO	ENAME	JOB	MGR	SAL	COMMISSION
101	Stonebridge	Data Analyst	10	100000.60	5000.90
102	Scott	Software developer	100	70000.96	3000.76
103	Crawford	Analyst	90	50000.00	3000.54
104	John	web developer	70	90000.00	53000.64
105	Collinson	HR	80	200000.00	7000.91

3. Update the column details of job

Update set job = 'Software developer' where empno = 105;

select * from employee;

4. Rename the column of Employee table using alter command.

~~alter table employee rename column set to salary;~~

select * from employee;

EMPNO	ENAME	JOB	MGR	SALARY	COMMISSION
101	Stonebridge	Data-Analyst	10	100000.60	5000.90
102	Scott	Software developer	100	70000.96	3000.76
103	Crowford	Analyst	90	50000.00	3000.54
104	John	web developer	70	90000.00	5000.64
105	Collinson	Software developer	80	200000.00	7000.91

EMPNO	ENAME	JOB	MGR	SALARY	COMMISSION
101	Stonebridge	Data-Analyst	10	100000.60	5000.90
102	Scott	Software developer	100	70000.96	3000.76
103	Crowford	Analyst	90	50000.00	3000.54
104	John	web developer	70	90000.00	5000.64
105	Collinson	Software developer	80	200000.00	7000.91

DATE 27/05/2024
EXP. NO. 2

EXPT TITLE: Queries using DDL & DML Commands

PAGE NO.

7

5 Delete the employee where empno = 105

delete employee where empno = 105;

select * from employee;

25/10

10

How affected

EMPNO	ENAME	JOB	HOUR	SALARY	COMMISSION
101	Stonebridge	Data-Analyst	100	100000.60	5000.96
102	Scott	Software-devel	100	70000.96	3000.76
103	Crowford	Analyst	90	50000.00	3000.54
104	Tonn	Webdeveloper	70	90000.00	5000.64

Queries using Aggregate functions (COUNT, AVG, MAX, MIN, SUM)

group by, order by; Employee (E-id, E-name, Age, Salary)

1. Create employee table containing all records E-id, E-name, Age, salary
2. Count no employee names from employee table.
3. Find the maximum age from employee table.
4. Find the minimum age from employee table.
5. Find the salaries of employee in ascending order.
6. Find the grouped Salaries of employees

1. Create employee table

Create employee table (empid number(4), e.name Varchar(20), Age(2),
Salary number(8,2));

close employee

insert into employee values (1, 'Stonebridge', 35, 100000.00);

insert into employee values (2, 'Scott', 39, 90000.96);

insert into employee values (3, 'Grawford', 45, 70000.54);

insert into employee values (4, 'Collinson', 47, 70.000.56);

Table	Column	Datatype	length	Precision	Scale	Primary key	Nullable	default	Commission
Employee	EID	Number	-	4	0	-	✓	-	-
	ENAME	Varchar	20	-	-	-	✓	-	-
	AGE	Number	-	2	0	-	✓	-	-
	SALARY	Number	-	6	2	-	✓	-	-
1-4									

1 row inserted

1 row inserted

1 row inserted

1 row inserted

select * from employee;

2. Count the number of employee names from employee table

select count (e.name) from employee;

3. Find the maximum age from employee table

select max(age) from employee;

4. Find the minimum age from employee table

select min(age) from employee;

EID	ENAME	AGE	SALARY
1	Stoneboudge	35	100000.00
2	Scott	39	90000.96
3	Crowyard	45	70000.56
4	Colinson	47	80000.56

COUNT (E.NAME)
4

MAX (AGE)
47

MIN (AGE)
35

5. Find the salaries of the employee in ascending order

select * from employee order by salary;

6. Find the grouped salaries of employees

select salary, count(salary) from employee group by salary;

15/25

10

EID	E_NAME	AGE	SALARY
3	Grawford	45	70000.56
4	Collinson	47	70000.56
2	Scott	39	90000.96
1	Stonebridge	35	100000.00

SALARY	COUNT(SALARY)
100000.00	1
90000.96	1
70000.56	2

Program - 1

Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference b/w the old or new salary.

CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)

-- Create the CUSTOMERS table

Create table CUSTOMERS (ID int Primary key, NAME Varchar (100), AGE int, ADDRESS Varchar (255), SALARY decimal (10,2));

- Create the trigger

Create or replace Trigger salary.difference.trigger before insert or update or delete on CUSTOMERS for each row
~~delete~~

old-salary NUMBER;

new-salary NUMBER;

begin

if inserting or updating then

old-salary := NVL (:OLD SALARY, 0);

new-salary := NVL (:NEW SALARY, 0);

DBMS_OUTPUT.PUT_LINE ('Salary difference : ' || (new-salary - old-salary));

else deleting then

old-salary := NVL (:OLD-SALARY, 0);

DBMS_OUTPUT.PUT_LINE ('Salary before deletion : ' || old-salary);

END IF;

Output

Table	column	Datatype	length	Precision	Scale	Primary key	Nullable	Default	Commission
CUSTOMERS	ID	NUMBERS	22	-	0	1	-	-	-
	NAME	VARCHAR	100	-	-	-	✓	-	-
	AGE	NUMBERS	22	-	0	-	✓	-	-
	ADDRESS	VARCHAR	255	-	-	-	✓	-	-
	SALARY	NUMBERS	-	10	2	-	✓	-	-

ii) Trigger created

iii) Salary difference 150000

1 row(s) inserted

ID	NAME	AGE	ADDRESS	SALARY
1	Stonebridge	24	123 main st	50000

Salary difference : 60000

ID	NAME	AGE	ADDRESS	SALARY
1	Stonebridge	24	123 main st	50000
2	Scott	23	23 main st	60000

Salary before deletion : 50000

1 row(s) deleted

ID	NAME	AGE	ADDRESS	SALARY
2	Scott	23	23 main st	60000

Salary difference : ~~630000~~

1 row(s) updated

ID	NAME	AGE	ADDRESS	SALARY
2	Scott	23	23 main st	70000

Program-5

Create curs cursor for employee table & extract the values from the table.
Declare the variables open the cursor & extract the values from the
cursor. close the cursor Employee [E-id, E-name, Age, salary]

Create table employee (E-id int Primarykey, E-name varchar(255), age int,
salary decimal(10,2));

- Declare variables

DECLARE

E-id Employee, E-id %Type;

E-name Employee, E-name %type;

Age Employee, E-name Age %Type;

Salary Employee, Salary %Type;

-- Declare cursor

~~cursor employee.cursor is~~

2 Select E-id, E-name, Age, salary;
from employee;

-- Open employee cursor;

-- Fetch data from cursor

loop

FETCH employee cursor INTO E-id, E-name, Age, salary;

EXIT WHEN employee cursor % NOT FOUND;

-- OUTPUT or use the fetched value

DBMS_OUTPUT.PUT_LINE ('Employee ID: ' || E-id || ', NAME: ' || E-name || ', Age: ' || Age || ', salary: ' || salary);

Desc employee;

Table	column	Datatype	length	Precision	Scale	Primary key	Nullable	Default	Commission
Employee	E-id	Number	22	-	0	-	✓	-	-
	E.name	Varchar	255	-	-	-	✓	-	-
	Age	NUMBER	22	-	0	-	✓	-	-
	SALARY	NUMBER	-	10	2	-	✓	-	-

DATE 11/7/24

EXPT TITLE: Create and cursor for employee table &
extract the values from the table declare the variables

EXP. NO. 05

PAGE NO.

13

END LOOP;

-- close the cursor

close employee.cursor;

END;

insert into employee values (001, 'Stonebridge', 20, 50000)

insert into employee values (002, 'Scott', 21, 90000)

~~25/12~~

10

Output:-

Employee ID : 1, Name : Stoneridge, Age : 20, Salary : 180000

Employee ID : 2, Name : Scott, Age : 21, Salary : 90600

Statement Processed.

Program-06

Write a PL/SQL block of code using Parameterize cursor, that will merge the data available in the newly created N-Roll call with the data available in the table O-Roll call with. If the data in the first table already exist in the second table then that data should be skipped.

Declare

V-count NUMBER;

CURSOR C.new-rollcall IS

SELECT id, name, roll

FROM N-rollcall;

BEGIN

FOR new-rec IN C.new-rollcall LOOP

-- check if the record already exists in O-Rollcall.

SELECT COUNT(*)

INTO V-count

FROM O-rollcall

WHERE id = new-rec.id;

-- if record doesn't exist insert it

IF V-count = 0 THEN

INSERT INTO O-rollcall (id, name, roll)

VALUES (new-rec.id, new-rec.name, new-rec.roll);

DBMS-OUTPUT.PUT_LINE('Record skipped: ' || new-rec.id);

END IF;

END LOOP;

OUTPUT:

select * from n-noucau2;

id	name	roll
1121	Satya	1234

select * from o-noucau2;

id	name	roll
1121	Satya	1234

select * from n-noucau4;

id	name	roll
1234	Scott	5678

record skipped : 1121

record inserted : 1234

DATE 15/7/2024

EXP. NO. 06

EXPT TITLE: Create two table to skip & insert the values

PAGE NO. 15

COMMIT;

END;

Select * from N-Moucaul2;

Select * from O-Moucaul2;

Insert INTO N-Moucaul2 (id, name, mou)

Values(1121, 'Satya', 12333, 11111111);

delete from O-Moucaul2 where id=121

25/7

10

Program-1

Install an open source code no sql database mongoDB & perform basic CRUD (create, read, update, delete)

Execute mongoDB basic queries using CRUD operation

Set Environment Path

Mongod - version

Mongosh

Use dbname

db.createCollection("collection name")

db.sama.insert({"name": "Alice", "Age": 30})

db.sama.find()

db.sama.update({"name": "Alice", \$set: "age": 31})

25/7

10

Output

mongosh

use kssem

Switched to db.kssem

```
db.kssem.insert([{"name": "Alice", "age": 30}])
```

```
db.kssem.find()
```

name: "Alice"

Age: 30

```
db.kssem.insert([{"Name": "Alice", "Age": 30, "Address": "Bangalore"}])
```

```
db.find()
```

name: "Alice"

age: 30

address: "Bangalore"