Teradata database and user management.

Create two databases as of below

1. Name: SCOTT\_DB
   1. PERMSPACE 20MB
   2. SPOOL SPACE 20MB
2. NAME: JAMES\_DB
   1. PERMSPACE 30MB
   2. SPOOL SPACE 20MB

Now create a use called SCOTT and JAMES for each database and give access to their specific databases

And create table as below IN SCOTT\_DB

STUDENT

ROLLNO INTERGER UPI

NAME VARCHAR NOT NULL

ADDRESS VARCHAR NOT NULL

TELNO VARCHAR UNIQUE

COURSE

COURSEID INTEGER UPI

COURSENAME VARCHAR NOT NULL

FEES INTEGER SHOULD BE MORE THEN 1000

COURSE\_REG

REGID INTEGER UPI

DATEOFREG DATE NOT NULL CANNOT BE FUTURE DATE

ROLLNO INTEGER FK OF STUDENT

COURSEID INTEGER FK OF COURSE

EXAM\_MARKS

EXAMID INTEGER UPI

DATEOFEXAM DATE NOT NULL

MARKS INTEGER CANNOT BE LESS THEN 0

FEES\_COLLECTED

RECEIPTNO INTEGER UPI

ROLLNO INTEGER FK TO STUDENT ROLLNO

FEES INTEGER SHOULD BE BASED OF REGISTRATION ID

REGID INTEGER FK COURSE\_REG TABLE

Data management

1. Every student on registration selects one course and pays a fee at the time of registration. So, for make the better data management the is needing to create a macro for inserting the data into the respective tables.

create macro insal (ROLLNO integer,NAME VARCHAR(20),ADDRESS VARCHAR(20),TELNO VARCHAR(20)

,COURSEID INTEGER,COURSENAME VARCHAR(20),FEES INTEGER,

REGID INTEGER ,DATEOFREG DATE,ROLL INTEGER,Cid INTEGER,

EXAMID INTEGER,DATEOFEXAM DATE,MARKS INTEGER

,RECEIPTNO INTEGER,Rn INTEGER,FEE INTEGER,RID INTEGER) AS

(

insert into student values (:ROLLNO,:NAME,:ADDRESS,:TELLNO);

insert into COURSE values (:COURSEID,:COURSENAME,:FEES);

insert into COURSE\_REG values (:REGID,:DATEOFREG,:ROLL,:Cid);

insert into EXAM\_MARKS values (:EXAMID,:DATEOFEXAM,:MARKS);

insert into FEES\_COLLECTED values (:RECEIPTNO,:Rn,:FEE,:RID);

)

1. After completing the exam, the same student can register for one more course. While registration for another course he/she should pay the fees. So according to the requirement please create a macro.

Create macro addstud as (id,dor,rno,cno)

(if(marks>50)

Then exec inscourse(:id,:dor,:rno,:cno)

)

Create macro inscourse (rid integer,dor date,rno integer,cno integer) as

(

insert into course\_reg(:rid,:dor,:rno,:cno);

)

1. Create a macro the address or telno change on the request of the student.

Data Reports

1. James want to access the some of the data from SCOTT\_DB like EXAM\_MARKS, STUDENT So, give Him access and check if he can write select statement on it.

 GRANT SELECT

     ON scott\_db.exam\_marks , scott\_db.student  
     TO james;

1. James wants to know what total fees is collected course wise from the database, so create an appropriate view and give him access.

Create view v1 as (select sum(fee) from FEES\_COLLECTED f, course\_reg c where f.regid=c.regid)

1. James want to develop two views for displaying the for PASSED\_STIUDENTS and FAILED\_STUDENTS (passing marks 50) check if he can do the same if not give him permission accordingly
2. James has appointed a new subordinate as CLARK. As required create a new user under the James for CLARK with 5MB of PERM SPACE and 5MB of SPOOL SPACE
3. Grant permission like SELECT, INSERT, UPDATE on STUDENT AND EXAM MARKS to Clark.
4. Clark can insert and update the marks of all the student, develop a macro for his daily use.

‘

CREATE SET TABLE SCOTT\_DB.STUDENT, FALLBACK

(

ROLLNO INTEGER,

NAME VARCHAR(20) NOT NULL,

ADDRESS VARCHAR(30) NOT NULL,

TELNO VARCHAR(10) UNIQUE NOT NULL

)

UNIQUE PRIMARY INDEX(ROLLNO);

CREATE SET TABLE COURSE

(

COURSEID INTEGER,

COURSENAME VARCHAR(20) NOT NULL,

FEES INTEGER NOT NULL, CHECK(FEES>1000)

)

UNIQUE PRIMARY INDEX(COURSEID);

CREATE SET TABLE SCOTT\_DB.COURSE\_REG, FALLBACK

(

REGID INTEGER,

DATEOFREG DATE ,

ROLLNO INTEGER,

COURSEID INTEGER,

FOREIGN KEY (ROLLNO) REFERENCES STUDENT(ROLLNO),

FOREIGN KEY (COURSEID) REFERENCES COURSE(COURSEID)

)

UNIQUE PRIMARY INDEX(REGID);

CREATE SET TABLE EXAM\_MARKS, FALLBACK

(

EXAMID INTEGER,

DATEOFEXAM DATE ,

MARKS INTEGER,

CHECK (MARKS>0)

)