Dr.Yasser Omar

Eng. Abdelrahman Saad

A room full of desks

Description automatically generated with medium confidence

School management system

Group members

1. mohamed Amr 20106990
2. Ziad Amr 20106000
3. mohamed Hany 20106887
4. Fady Ayman 20106800

School description:

Our school project targets some points one of them is that the manager wants to know some information about student for example(email ,gender, mobile ,date of birth ,name ,current level, class name)and the system should differentiate between students and there is other information about student’s parent for example (national id , job ,name ,address ,mobile ,mobile) also other information that the student depends on his parent and a parent may have more than one child and we should include that point in our system. The second information is that system should store teacher information like (salary ,gender ,mobile, national id, birth of date, address ,name ,email) and a teacher can teach only one course and system must track course and which teacher is teaching that course and the system must include that point and the system should include course attributes for example name and for example the manager wants to know how many student is taking physics or other course so the system should handle that part in addition to the school have only 8 transports for all students and the student can subscribe into that transport by paying more fees and its optional to use school transport and the system have 2 types of registration one is online and other is physical and the system have 2 types of registration one for students and the pay and one for employees and their information and the system have 3 types of employees and they are drivers , teachers and other employee for department the system have 4 normal employees work for 1 department and the school have 8 drivers work on the 8 transport and student can have more than one course and the system must tract students grades and the manager wants to know any updating steps happen to students grades to know who update that data to know is that student really deserve to change its grade or not deserve.

Rough Schema

A picture containing text, black, control

Description automatically generated

Final Schema

Diagram

Description automatically generated

1m

1m

1

1m

1m

mm

1m

mm

1m

1m

1m

mm

1dm

mm

1m

1

1m

1m

mm

1m

1m

mm

mm

mm

Mapping :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| sid | email | gender | mobile | DOB | Current\_level | class | address | SR\_id | name | T\_no |

Student

FK

FK

Primary key : Sid

Sid : student id

DOB: Date of Birth

SR\_id: Student Registration

T\_id: Transport \_no

Course

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| code | level | Name | Dep\_id | T\_id |

primary key : code

FK

FK

Dep\_id : Department ID

T\_id : Teacher\_id

Department

|  |  |
| --- | --- |
| Dep\_id | Name |

Primary Key : dep\_id

Dep\_id : Department id

Student Registration

|  |  |  |  |
| --- | --- | --- | --- |
| SR\_id | date | amount | T\_code |

Primary key : SR\_id

FK

T\_code : type code

It refers to online or physical operation

Type (type of registration)

|  |  |
| --- | --- |
| *T\_code* | Name |

Primary Key : T\_code

T\_code : type code

It refers to online or physical operation

Employee Registration

|  |  |  |  |
| --- | --- | --- | --- |
| *ER\_id* | date | T\_id | Driver\_id |

Primary key : ER\_id

FK

FK

ER\_id : Employee Registration id

T\_id : Teacher id

That table takes only one employee

Teacher

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *T\_id* | National\_id | name | DOB | mobile | salary | address |

Primary key : T\_id

DOB: Date of Birth

Parent

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sid | Pid | National\_id | name | mobile | address | Job |

FK

Primary key : (sid,pid)

Sid : Student id

Pid : Parent id

Enrollment

|  |  |  |
| --- | --- | --- |
| *Sid*  FK | code | grade |

FK

Primary Key : (Sid,code)

Transport

|  |  |
| --- | --- |
| T\_no | Driver\_id |

FK

Primary Key : T\_no

T\_no : Transport Number

Driver

|  |  |  |  |
| --- | --- | --- | --- |
| Driver id | name | Salary | National id |

Primary Key : Driver id

Department Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| emp id | Dep id | name | Mobile | salary |

FK

Creating tables

CREATE TABLE student (

sid NUMBER primary key,

std\_id number,

trans\_id number,

name VARCHAR2(100) ,

email VARCHAR2(100),

mobile VARCHAR2(12),

gender VARCHAR2(10),

data\_of\_birth date,

current\_level NUMBER,

class\_name VARCHAR2(10),

address VARCHAR2(100)

);

CREATE TABLE student\_registration (

std\_id NUMBER,

stud\_registration date,

type\_code number,

amount NUMBER NOT NULL,

PRIMARY KEY (std\_id)

);

CREATE TABLE Rtype (

Type\_code NUMBER primary key,

Type\_name VARCHAR2(15)

);

CREATE TABLE transport (

trans\_id NUMBER primary key,

driver\_id number

);

CREATE TABLE department (

dep\_id NUMBER primary key,

name VARCHAR2(100)

);

CREATE TABLE course (

code NUMBER,

name VARCHAR2(100),

course\_level NUMBER,

dep\_id NUMBER,

teacher\_id NUMBER,

PRIMARY KEY (code)

);

CREATE TABLE teacher (

teacher\_id NUMBER primary key,

name VARCHAR2(100) ,

national\_id VARCHAR2(14),

age number,

mobile VARCHAR2(12),

salary NUMBER,

gender VARCHAR2(10)

);

CREATE TABLE employee\_registration (

emp\_register\_id NUMBER,

emp\_registration date,

driver\_id NUMBER,

teacher\_id NUMBER,

primary key (emp\_register\_id)

);

CREATE TABLE driver (

driver\_id NUMBER primary key,

national\_id varchar2(14),

name VARCHAR2(100),

salary NUMBER

);

CREATE TABLE enrollment (

sid NUMBER,

code NUMBER,

grade VARCHAR2(10),

PRIMARY KEY (sid,code)

);

create table dep\_emp(

emp\_id number primary key,

dep\_id number,

name varchar2(50),

mobile varchar2(12),

salary number

);

create table parent (

pid number,

sid number,

name varchar2(100),

national\_id varchar2(14),

mobile varchar2(12),

job varchar2(100),

address varchar2(100),

primary key (pid,sid)

);

Alter Statements for Adding FOREIGN KEY

alter table enrollment add FOREIGN KEY (sid) REFERENCES student (sid);

alter table enrollment add FOREIGN KEY (code) REFERENCES course (code);

alter table employee\_registration add FOREIGN KEY (teacher\_id) REFERENCES teacher (teacher\_id);

alter table employee\_registration add FOREIGN KEY (driver\_id) REFERENCES driver (driver\_id);

alter table parent add foreign key (sid) references student(sid);

alter table student add foreign key (std\_id) references student\_registration(std\_id);

alter table student add foreign key (trans\_id) references transport (trans\_id);

alter table student\_registration add foreign key (type\_code) references Rtype (type\_code);

alter table course add foreign key (teacher\_id) references teacher (teacher\_id);

alter table transport add foreign key (driver\_id) references driver (driver\_id);

alter table course add FOREIGN KEY (dep\_id) REFERENCES department (dep\_id);

alter table dep\_emp add FOREIGN KEY (dep\_id) REFERENCES department (dep\_id);

Examples for INSERTION

**insert into driver values(1,'12345678945612','ahmed aly',3000);**

**insert into transport values (1,1);**

**insert into teacher values(1,'emad abdallah','30145698762145',25,'01270804568',5000,'male');**

**insert into employee\_registration (emp\_register\_id,emp\_registration,driver\_id) values(1,to\_date('01-FEB-1996','DD-MON-YYYY'),1);**

**insert into Rtype values(1,'physical');**

**insert into department values (1,'Admission');**

**insert into student\_registration values(1,to\_date('01-aug-2010','DD-MON-YYYY'),1,4000);**

**insert into student values (1,1,1,'ziad amr','ziad11@gmail.com','01270931560','male',to\_date('12-jun-2005','DD-MON-YYYY'),10,'10A','mokatam 81 street');**

**insert into course values (1,'physics',10,1,1);**

**insert into enrollment (sid,code,grade) values (1,2,'A');**

**insert into parent values (1,1,'amr ahmed','12365478910548','01061075423','doctor','mokatam 81 street');**

-- queries to help us in analysis

**select \* from student where sid = 1 OR sid= 2;**

**select \* from student\_registration where std\_id = 1 or std\_id=2;**

**select \* from teacher where teacher\_id = 1 or teacher\_id= 2;**

**select \* from enrollment where sid = 1 and sid=2;**

**select \* from course where code= 1 or code=2;**

**select \* from driver;**

**select \* from transport;**

complex query

first query

**select \* from course**

**inner join teacher**

**ON course.teacher\_id = teacher.teacher\_id;**

Second query

**select name,enrollment.sid,count(code) As "Number Of Courses"**

**from student**

**inner join enrollment**

**on student.sid=enrollment.sid**

**group by enrollment.sid,name**

**order by count(code) asc;**

third query

**select teacher.name ,course.name**

**from teacher**

**inner join course on**

**teacher.teacher\_id = course.code;**

fourth query

**select S.sid, S.name AS Student\_Name ,C.name AS Course\_Name, E.grade AS Student\_grade ,T.name AS teacher\_Name**

**from enrollment E**

**inner join student S on E.sid = S.sid**

**inner join course C on E.code = C.code**

**inner join teacher T on C.teacher\_id = T.teacher\_id;**

fifth query

**select name "Parent Name", count(sid) AS "Number Of Childern " from parent group by name;**

sixth query

**select S.name as StudentName,S.CURRENT\_LEVEL,D.name As DriverName,SR.ammount,RT.type\_name AS Registration\_Type**

**from student S, student\_registration SR,transport T,driver D,Rtype RT**

**where S.sid = SR.std\_id**

**and S.trans\_id = T.trans\_id**

**and T.driver\_id = D.driver\_id**

**and SR.type\_code=RT.type\_code;**

triggers

**create table auditGrade (sid number, oldgrade varchar2(10) , newgrade varchar2(10) , userr varchar2(40) , trxdate date)**

**create or replace trigger lockGrade before update of grade on enrollment**

**for each row**

**declare**

**begin**

**insert into auditGrade(sid,oldgrade,newgrade,userr,trxdate)**

**values(:new.sid,:old.grade,:new.grade,user,sysdate);**

**end;**

**update enrollment set grade='A+' where sid=1;**

**select \* from auditGrade;**

function For Straight A+

create or replace function ChkStraightA(p\_sid number)

return varchar2

is

v\_count number;

begin

select count(\*)

into v\_count

from enrollment

where sid=p\_sid

and grade <> 'A+';

if v\_count =0

then return 'Straight A+';

else return NULL;

end if;

end;

select sid,name,ChkStraightA(sid) AS "Stright A+" from student;