

Ayush Goyal

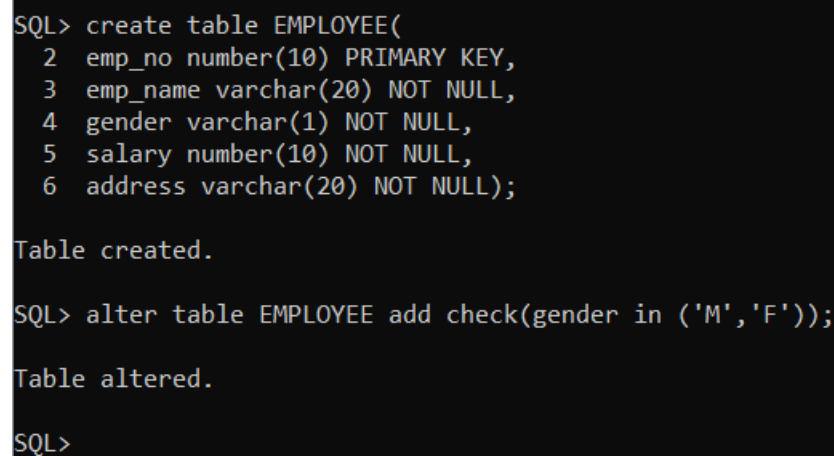
190905522 CSE D 62

DBS LAB 2 (Week 2) – Integrity Constraints

Q1) Create Employee table with following constraints:•Make EmpNo as Primary key.•Do not allow EmpName, Gender, Salary and Address to have null values.•Allow Genderto have one of the two values: 'M', 'F'.

```
create table EMPLOYEE(  
emp_no number(10) PRIMARY KEY,  
emp_name varchar(20) NOT NULL,  
gender varchar(1) NOT NULL,  
salary number(10) NOT NULL,  
address varchar(20) NOT NULL);
```

```
alter table EMPLOYEE add check(gender in ('M','F'));
```

A screenshot of a terminal window showing the execution of SQL commands. The first command creates the EMPLOYEE table with columns emp_no (primary key), emp_name, gender, salary, and address. The second command adds a check constraint to the gender column, allowing only 'M' or 'F'. The terminal output shows 'Table created.' and 'Table altered.'.

```
SQL> create table EMPLOYEE(  
2 emp_no number(10) PRIMARY KEY,  
3 emp_name varchar(20) NOT NULL,  
4 gender varchar(1) NOT NULL,  
5 salary number(10) NOT NULL,  
6 address varchar(20) NOT NULL);  
  
Table created.  
  
SQL> alter table EMPLOYEE add check(gender in ('M','F'));  
  
Table altered.  
  
SQL>
```

Q2)Create Department table with following:•Make DeptNo as Primary key•Make DeptName as candidate key

```
create table DEPARTMENT(  
dept_no number(10) PRIMARY KEY,  
dept_name varchar(20) NOT NULL UNIQUE,  
location varchar(20));
```

```
SQL> create table DEPARTMENT(  
  2  dept_no number(10) PRIMARY KEY,  
  3  dept_name varchar(20) NOT NULL UNIQUE,  
  4  location varchar(20));
```

Table created.

```
SQL>
```

Q3) Make DNo of Employee as foreign key which refers to DeptNo of Department

```
alter table EMPLOYEE add(dept_no number(10));
```

```
alter table EMPLOYEE add FOREIGN KEY(dept_no) references DEPARTMENT(dept_no);
```

```
SQL> alter table EMPLOYEE add(dept_no number(10));
```

Table altered.

```
SQL> alter table EMPLOYEE add FOREIGN KEY(dept_no) references DEPARTMENT(dept_no);
```

Table altered.

```
SQL>
```

Q4) Insert few tuples into Employee and Department which satisfies the above constraints.

```
insert into DEPARTMENT(dept_no,dept_name,location) values(101,'Worker','Hyderabad');
```

```
insert into DEPARTMENT(dept_no,dept_name,location) values(102,'Ayush','Kolkata');
```

```
insert into EMPLOYEE(emp_no,emp_name,gender,salary,address,dept_no)  
values(1101,'Dipesh','M',1000,'Hyderabad',101);
```

```
insert into EMPLOYEE(emp_no,emp_name,gender,salary,address,dept_no)  
values(1102,'AG','M',2000,'Manipal',102);
```

```
SQL> insert into DEPARTMENT(dept_no,dept_name,location) values(101,'Worker','Hyderabad');
```

1 row created.

```
SQL> insert into DEPARTMENT(dept_no,dept_name,location) values(102,'Ayush','Kolkata');
```

1 row created.

```
SQL> insert into EMPLOYEE(emp_no,emp_name,gender,salary,address,dept_no) values(1101,'Dipesh','M',1000,'Hyderabad',101);
```

1 row created.

```
SQL> insert into EMPLOYEE(emp_no,emp_name,gender,salary,address,dept_no) values(1102,'AG','M',2000,'Manipal',102);
```

Q5) Try to insert few tuples into Employee and Department which violates some of the above constraints.

```
insert into DEPARTMENT(dept_no,dept_name,location) values(101,'Woek','Hyderabad');
```

```
insert into DEPARTMENT(dept_no,dept_name,location) values(102,'Polu','Manipal');
```

```
insert into EMPLOYEE(emp_no,emp_name,gender,salary,address,dept_no)
values(001,'DipeshSingh','M',500,'Hyderabad',101);
```

```
insert into EMPLOYEE(emp_no,emp_name,gender,salary,address,dept_no)
values(002,'AG','M',1000,'Manipal',102);
```

```
SQL> insert into DEPARTMENT(dept_no,dept_name,location) values(101,'Woek','Hyderabad');
insert into DEPARTMENT(dept_no,dept_name,location) values(101,'Woek','Hyderabad')
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.SYS_C007029) violated

SQL> insert into DEPARTMENT(dept_no,dept_name,location) values(102,'Polu','Manipal');
insert into DEPARTMENT(dept_no,dept_name,location) values(102,'Polu','Manipal')
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.SYS_C007029) violated

SQL> insert into EMPLOYEE(emp_no,emp_name,gender,salary,address,dept_no) values(001,'DipeshSingh','M',500,'Hyderabad',101);
1 row created.

SQL>
```

Q6) Try to modify/delete a tuple which violates a constraint.

```
alter table EMPLOYEE drop constraint abc;
```

```
delete from DEPARTMENT where dept_no='101';
```

```
SQL> alter table EMPLOYEE drop constraint abc;
alter table EMPLOYEE drop constraint abc
*
ERROR at line 1:
ORA-02443: Cannot drop constraint - nonexistent constraint
```

```
SQL> delete from DEPARTMENT where dept_no = '101';
delete from DEPARTMENT where dept_no = '101'
*
ERROR at line 1:
ORA-02292: integrity constraint (SYSTEM.SYS_C007031) violated - child record found
```

Q7) Modify the foreign key constraint of Employee table such that whenever a department tuple is deleted, the employees belonging to that department will also be deleted.

```
alter table EMPLOYEE drop constraint abc;
```

```
alter table Employee drop constraint SYS_C007031;
```

```
select constraint_name, table_name, constraint_type from user_constraints;
```

```
alter table Employee add constraint FK foreign key(Dno) references Department(DeptNo) on delete cascade;
```

```
CONSTRAINT_NAME          TABLE_NAME                C
-----
SYS_C007023               EMPLOYEE                   C
SYS_C007024               EMPLOYEE                   C
SYS_C007025               EMPLOYEE                   C
SYS_C007026               EMPLOYEE                   P
SYS_C007027               EMPLOYEE                   C
SYS_C007028               DEPARTMENT                 C
SYS_C007029               DEPARTMENT                 P
SYS_C007030               DEPARTMENT                 U
SYS_C007031               EMPLOYEE                   R

460 rows selected.

SQL> alter table Employee drop constraint SYS_C007031;

Table altered.
```

```
SQL> alter table Employee add constraint FK foreign key(dept_no) references Department(dept_no) on delete cascade;

Table altered.

SQL>
```

Q8) Create a named constraint to set the default salary to 10000 and test the constraint by inserting a new record.

```
alter table EMPLOYEE modify(salary default 10000);
```

```
SQL> alter table EMPLOYEE modify(salary default 10000);

Table altered.

SQL>
```

UNIVERSITY DATA BASE ADDED

@ "D:\CSE\DBS Lab\DDL+drop"

@ "D:\CSE\DBS Lab\smallRelationsInsertFile"

This imports all tables and populates them.

Q9) List all Students with names and their department names.

Select name,dept_name from student;

```
SQL> Select name,dept_name from student;
```

NAME	DEPT_NAME
Zhang	Comp. Sci.
Shankar	Comp. Sci.
Brandt	History
Chavez	Finance
Peltier	Physics
Levy	Physics
Williams	Comp. Sci.
Sanchez	Music
Snow	Physics
Brown	Comp. Sci.
Aoi	Elec. Eng.
Bourikas	Elec. Eng.
Tanaka	Biology

13 rows selected.

Q10) List all instructors in CSE department.

Select name,dept_name from instructor where dept_name='Comp. Sci.';

```
SQL> Select name,dept_name from instructor where dept_name='Comp. Sci.';
```

NAME	DEPT_NAME
Srinivasan	Comp. Sci.
Katz	Comp. Sci.
Brandt	Comp. Sci.

Q11) Find the names of courses in CSE department which have 3 credits.

Select title,credits from course where dept_name='Comp. Sci.' and credits=3;

```
SQL> Select title,credits from course where dept_name='Comp. Sci.' and credits=3;
```

TITLE	CREDITS
Robotics	3
Image Processing	3
Database System Concepts	3

Q12) For the student with ID 12345 (or any other value), show all course-id and title of all courses registered for by the student.

Select title,course_id from course natural join takes where ID=12345;

```
SQL> Select title,course_id from course natural join takes where ID=12345;
```

TITLE	COURSE_I
Intro. to Computer Science	CS-101
Game Design	CS-190
Robotics	CS-315
Database System Concepts	CS-347

Q13) List all the instructors whose salary is in between 40000 and 90000.

Select name from instructor where salary between 40000 and 90000;

```
SQL> Select name from instructor where salary between 40000 and 90000;
```

NAME
Srinivasan
Wu
Mozart
El Said
Gold
Katz
Califieri
Singh
Crick
Kim

10 rows selected.

```
SQL>
```

Q14) Display the IDs of all instructors who have never taught a course.

Select instructor.id from instructor where id not in (select distinct teaches.id from teaches);

```
SQL> Select instructor.id from instructor where id not in (select distinct teaches.id from teaches);
```

ID
33456
58583
76543

Q15) Find the student names, course names, and the year, for all students those who have attended classes in room-number 303.

Select name, title, takes.year FROM student, section, course, takes WHERE room_number=3128 and course.course_id = section.course_id and course.course_id = takes.course_id and takes.id = student.id and takes.year = section.year and takes.sec_id = section.sec_id and section.semester = takes.semester;

```
SQL> Select name, title, takes.year FROM student, section, course, takes WHERE room_number=3128 and course.course_id = section.course_id and course.course_id = takes.course_id and takes.id = student.id and takes.year = section.year and takes.sec_id = section.sec_id and section.semester = takes.semester;
```

NAME	TITLE
Shankar	Game Design
Williams	Game Design
Brown	Image Processing
Zhang	Database System Concepts
Shankar	Database System Concepts
Aoi	Intro. to Digital Systems

6 rows selected.

Room 303 has no students:

```
SQL> Select name, title, takes.year FROM student, section, course, takes WHERE room_number=303 and course.course_id = section.course_id and course.course_id = takes.course_id and takes.id = student.id and takes.year = section.year and takes.sec_id = section.sec_id and section.semester = takes.semester;
```

no rows selected

16) For all students who have opted courses in 2015, find their names and course id's with the attribute course title replaced by c-name.

select name, course_id as c_name from student natural join takes where takes.year=2015;

(As we can see, 2009 has no rows)

```
SQL> select name, course_id as c_name from student natural join takes where takes.year=2009;

NAME                C_NAME
-----
Zhang                CS-101
Zhang                CS-347
Shankar              CS-101
Shankar              CS-190
Shankar              CS-347
Peltier              PHY-101
Levy                 CS-101
Williams             CS-101
Williams             CS-190
Brown                CS-101
Aoi                  EE-181

NAME                C_NAME
-----
Bourikas             CS-101
Tanaka                BIO-101

13 rows selected.

SQL> select name, course_id as c_name from student natural join takes where takes.year=2015;

no rows selected

SQL>
```

17) Find the names of all instructors whose salary is greater than the salary of at least one instructor of CSE department and salary replaced by inst-salary.

select distinct a.name, a.salary as inst_salary from instructor a, instructor b where b.dept_name='Comp. Sci.' and a.salary>b.salary;

```
SQL> select distinct a.name, a.salary as inst_salary from instructor a, instructor b where b.dept_name='Comp. Sci.' and a.salary>b.salary;

NAME                INST_SALARY
-----
Gold                 87000
Katz                 75000
Crick                72000
Einstein             95000
Wu                   90000
Brandt               92000
Singh                80000
Kim                  80000

8 rows selected.
```

18) Find the names of all instructors whose department name includes the substring 'ch'.

Select name from instructors where dept_name like '%ch%';

(As we can see no dept name has ch as substring)

```
SQL> Select name from instructor where dept_name like '%ch%';

no rows selected

SQL> Select name from instructor where dept_name like '%P%';

NAME
-----
Einstein
Gold

SQL>
```

19) List the student names along with the length of the student names.

Select name,LENGTH(name) from student;

```
SQL> Select name,LENGTH(name) from student;

NAME                LENGTH(NAME)
-----
Zhang                5
Shankar              7
Brandt              6
Chavez              6
Peltier             7
Levy                 4
Williams            8
Sanchez             7
Snow                4
Brown               5
Aoi                  3

NAME                LENGTH(NAME)
-----
Bourikas            8
Tanaka              6

13 rows selected.

SQL>
```

20) List the department names and 3 characters from 3rd position of each department name

select dept_name,substr(dept_name,3,3) from department;

```
SQL> select dept_name,substr(dept_name,3,3) from department;
```

DEPT_NAME	SUBSTR(DEPT_
-----	-----
Biology	olo
Comp. Sci.	mp.
Elec. Eng.	ec.
Finance	nan
History	sto
Music	sic
Physics	ysi

7 rows selected.

```
SQL>
```

21) List the instructor names in upper case.

Select UPPER(name) from instructor;

```
SQL> Select UPPER(name) from instructor;
```

UPPER(NAME)

SRINIVASAN
WU
MOZART
EINSTEIN
EL SAID
GOLD
KATZ
CALIFIERI
SINGH
CRICK
BRANDT

UPPER(NAME)

KIM

12 rows selected.

```
SQL>
```

22) Replace NULL with value1(say 0) for a column in any of the table

Select NVL(grade,'F') from takes;

```
SQL> Select NVL(grade,'F') from takes;
```

```
NV
--
A
A-
C
A
A
A
B
C+
B-
F
B+
```

```
NV
--
B
A-
B+
A-
A
A
C
C-
B
A
F
```

```
22 rows selected.
```

23) Display the salary and salary/3 rounded to nearest hundred from Instructor.

Select salary,ROUND(salary/3,-2) from instructor;

```
SQL> Select salary,ROUND(salary/3,-2) from instructor;
```

SALARY	ROUND(SALARY/3,-2)
65000	21700
90000	30000
40000	13300
95000	31700
60000	20000
87000	29000
75000	25000
62000	20700
80000	26700
72000	24000
92000	30700

SALARY	ROUND(SALARY/3,-2)
80000	26700

```
12 rows selected.
```

```
SQL>
```

THE END

