

```

SQL> /*
SQL> CIS 353 - Team 6
SQL>     Drew Van Maanen
SQL>     Joseph Coscarelli
SQL>     Andy Hung
SQL>     Brendon Murthum
SQL>     Carter Garety
SQL> */
SQL>
SQL> --
SQL> -- CREATING THE SCHEMA
SQL> --
SQL>
SQL> DROP TABLE club CASCADE CONSTRAINTS;

```

Table dropped.

```
SQL> DROP TABLE student CASCADE CONSTRAINTS;
```

Table dropped.

```
SQL> DROP TABLE teacher CASCADE CONSTRAINTS;
```

Table dropped.

```
SQL> DROP TABLE classes CASCADE CONSTRAINTS;
```

Table dropped.

```
SQL> DROP TABLE room CASCADE CONSTRAINTS;
```

Table dropped.

```
SQL> DROP TABLE member_of CASCADE CONSTRAINTS;
```

Table dropped.

```
SQL> DROP TABLE enrolled_in CASCADE CONSTRAINTS;
```

Table dropped.

```
SQL> DROP TABLE student_phone CASCADE CONSTRAINTS;
```

Table dropped.

```

SQL>
SQL> CREATE TABLE club(
  2   club_subject          char(15),
  3   club_meet_time        char(9)          NOT NULL,
  4   club_tid              char(2),
  5   CONSTRAINT club_IC_1 CHECK (club_meet_time IN
('monday','tuesday','wednesday','thursday','friday','saturday','sunday')),
  6   CONSTRAINT club_PK PRIMARY KEY (club_subject)
  7   DEFERRABLE INITIALLY DEFERRED
  8 );

```

Table created.

```

SQL>
SQL> CREATE TABLE student(
  2   student_ssn          char(9),
  3   student_name          char(30)          NOT NULL,
  4   student_bdate         date              NOT NULL,
  5   student_major         char(30),
  6   student_credit_hours  number(4),
  7   CONSTRAINT student_PK PRIMARY KEY (student_ssn)
  8   DEFERRABLE INITIALLY DEFERRED

```

```
9 );
```

Table created.

```
SQL>
```

```
SQL> CREATE TABLE teacher(  
2     teacher_id          char(2),  
3     teacher_name        char(30)    NOT NULL,  
4     CONSTRAINT teacher_PK PRIMARY KEY (teacher_id)  
5 );
```

Table created.

```
SQL>
```

```
SQL> CREATE TABLE classes(  
2     class_room_num      char(5),  
3     class_time          char(30),  
4     class_tid           char(2),  
5     class_course        char(30),  
6     CONSTRAINT classes_PK PRIMARY KEY (class_room_num, class_time)  
7     DEFERRABLE INITIALLY DEFERRED  
8 );
```

Table created.

```
SQL>
```

```
SQL> CREATE TABLE room (  
2     room_num            char(5),  
3     room_location       char(30)    NOT NULL,  
4     room_max_occupancy  number(3)    NOT NULL,  
5     room_projectors     number(2)    NOT NULL,  
6     CONSTRAINT room_PK PRIMARY KEY (room_num)  
7     DEFERRABLE INITIALLY DEFERRED,  
8     -- This says a room must have at least 30 people if it has 2 projectors  
9     CONSTRAINT room_IC_1 CHECK (NOT (room_max_occupancy < 30 AND  
room_projectors > 1))  
10 );
```

Table created.

```
SQL>
```

```
SQL> CREATE TABLE member_of (  
2     member_student_ssn  char(9),  
3     member_club_subject char(15),  
4     member_date_joined  date        NOT NULL,  
5     CONSTRAINT member_of_PK PRIMARY KEY (member_student_ssn,  
member_club_subject)  
6     DEFERRABLE INITIALLY DEFERRED  
7 );
```

Table created.

```
SQL>
```

```
SQL> CREATE TABLE enrolled_in (  
2     enrolled_student_ssn char(9),  
3     enrolled_class_room_num char(5),  
4     enrolled_class_time   char(30),  
5     CONSTRAINT enrolled_in_PK PRIMARY KEY (enrolled_student_ssn,  
enrolled_class_room_num, enrolled_class_time)  
6     DEFERRABLE INITIALLY DEFERRED  
7 );
```

Table created.

```
SQL>
```

```
SQL> CREATE TABLE student_phone (  
2     phone_student_ssn   char(9),
```

```
3     phone_number          char(10),
4     CONSTRAINT student_phone_PK PRIMARY KEY (phone_student_ssn, phone_number)
5 );
```

Table created.

```
SQL>
SQL> -- Add the foreign keys:
SQL> ALTER TABLE club
2  ADD CONSTRAINT club_FK_1 FOREIGN KEY (club_tid) REFERENCES teacher
  (teacher_id)
3      ON DELETE SET NULL
4      DEFERRABLE INITIALLY DEFERRED;
```

Table altered.

```
SQL>
SQL> ALTER TABLE classes
2  ADD CONSTRAINT classes_FK_1 FOREIGN KEY (class_tid) REFERENCES teacher
  (teacher_id)
3      ON DELETE CASCADE
4      DEFERRABLE INITIALLY DEFERRED;
```

Table altered.

```
SQL> ALTER TABLE classes
2  ADD CONSTRAINT classes_FK_2 FOREIGN KEY (class_room_num) REFERENCES room
  (room_num)
3      ON DELETE CASCADE
4      DEFERRABLE INITIALLY DEFERRED;
```

Table altered.

```
SQL>
SQL> ALTER TABLE member_of
2  ADD CONSTRAINT member_of_FK_1 FOREIGN KEY (member_student_ssn) REFERENCES
  student(student_ssn)
3      ON DELETE CASCADE
4      DEFERRABLE INITIALLY DEFERRED;
```

Table altered.

```
SQL> ALTER TABLE member_of
2  ADD CONSTRAINT member_of_FK_2 FOREIGN KEY (member_club_subject) REFERENCES
  club(club_subject)
3      ON DELETE CASCADE
4      DEFERRABLE INITIALLY DEFERRED;
```

Table altered.

```
SQL>
SQL> ALTER TABLE enrolled_in
2  ADD CONSTRAINT enrolled_in_FK_1 FOREIGN KEY (enrolled_student_ssn)
  REFERENCES student(student_ssn)
3      ON DELETE CASCADE
4      DEFERRABLE INITIALLY DEFERRED;
```

Table altered.

```
SQL> ALTER TABLE enrolled_in
2  ADD CONSTRAINT enrolled_in_FK_2 FOREIGN KEY (enrolled_class_room_num,
  enrolled_class_time) REFERENCES classes(class_room_num, class_time)
3      ON DELETE CASCADE
4      DEFERRABLE INITIALLY DEFERRED;
```

Table altered.

```

SQL>
SQL> ALTER TABLE student_phone
2 ADD CONSTRAINT student_phone_FK_1 FOREIGN KEY (phone_student_ssn)
REFERENCES student(student_ssn)
3 ON DELETE CASCADE
4 DEFERRABLE INITIALLY DEFERRED;

```

Table altered.

```

SQL>
SQL> --
SQL> -- POPULATING THE TABLES
SQL> --
SQL>
SQL> SET FEEDBACK OFF
SQL> alter session set      NLS_DATE_FORMAT = 'YYYY-MM-DD';
SQL>
SQL> --student(ssn, name, birthday, major, number of credit hours currently)
SQL> INSERT INTO student VALUES (100000001, 'Drew VM', '1989-08-12', 'computer
science', 60);
SQL> INSERT INTO student VALUES (100000002, 'Andy Hung', '1996-08-19', 'computer
science', 40);
SQL> INSERT INTO student VALUES (100000003, 'Joey Coscarelli', '1992-12-08',
'computer science', 50);
SQL> INSERT INTO student VALUES (100000004, 'Morgan Freeman', '1963-02-10',
'theatre', 10);
SQL> INSERT INTO student VALUES (100000005, 'Judi Dench', '1972-11-19',
'theatre', 26);
SQL> INSERT INTO student VALUES (100000006, 'Philip Philips', '1990-12-27',
'mechanical engineering', 31);
SQL>
SQL> --teacher(2 digit ID, name)
SQL> INSERT INTO teacher VALUES (13, 'Dave Wilson');
SQL> INSERT INTO teacher VALUES (12, 'Queen Takerook');
SQL> INSERT INTO teacher VALUES (11, 'Jane Doe');
SQL> INSERT INTO teacher VALUES (10, 'Huey Lewis');
SQL> INSERT INTO teacher VALUES (14, 'John Smith');
SQL> INSERT INTO teacher VALUES (15, 'Peter Parker');
SQL>
SQL> --club(name of club, lowercase day it meets, teacher ID who runs it)
SQL> INSERT INTO club VALUES ('physics', 'tuesday', 13);
SQL> INSERT INTO club VALUES ('chess', 'thursday', 12);
SQL> INSERT INTO club VALUES ('math', 'tuesday', 11);
SQL> INSERT INTO club VALUES ('news', 'monday', 10);
SQL>
SQL> --member_of(student ssn, name of club, date joined)
SQL> INSERT INTO member_of VALUES (100000001, 'physics', '2016-01-01');
SQL> INSERT INTO member_of VALUES (100000001, 'math', '2016-02-01');
SQL> INSERT INTO member_of VALUES (100000003, 'chess', '2015-02-14');
SQL> INSERT INTO member_of VALUES (100000004, 'chess', '2016-10-22');
SQL> INSERT INTO member_of VALUES (100000004, 'news', '2016-01-13');
SQL> INSERT INTO member_of VALUES (100000005, 'news', '2015-10-12');
SQL> INSERT INTO member_of VALUES (100000005, 'math', '2015-05-13');
SQL> INSERT INTO member_of VALUES (100000005, 'chess', '2014-05-13');
SQL> INSERT INTO member_of VALUES (100000005, 'physics', '2014-05-13');
SQL> INSERT INTO member_of VALUES (100000006, 'math', '2016-12-21');
SQL>
SQL> --student_phone(student ssn, phone number)<-multivalued attribute
SQL> INSERT INTO student_phone VALUES (100000001, '364-8901');
SQL> INSERT INTO student_phone VALUES (100000001, '364-1102');
SQL> INSERT INTO student_phone VALUES (100000003, '867-5309');
SQL> INSERT INTO student_phone VALUES (100000005, '312-7775');
SQL> INSERT INTO student_phone VALUES (100000005, '312-2150');
SQL> INSERT INTO student_phone VALUES (100000006, '616-1234');
SQL>
SQL> --room(fake room number must be <= 5, location, max occupancy, number of
projectors)

```

```

SQL> --check constraint room_IC_1
SQL> INSERT INTO room VALUES ('A117', 'mac', 35, 1);
SQL> INSERT INTO room VALUES ('D2117', 'mac', 40, 2);
SQL> INSERT INTO room VALUES ('C100', 'pad', 12, 0);
SQL> INSERT INTO room VALUES ('F2120', 'pac', 32, 0);
SQL> INSERT INTO room VALUES ('A1200', 'man', 36, 2);
SQL>
SQL> --classes(room, days it meets, teacher id, course name)
SQL> --days it meets = M, Tu, W, Th, F
SQL> INSERT INTO classes VALUES ('A117', 'MWF', 13, 'phy 220');
SQL> INSERT INTO classes VALUES ('A117', 'TuTh', 13, 'phy 220');
SQL> INSERT INTO classes VALUES ('C100', 'TuTh', 13, 'mth 325');
SQL> INSERT INTO classes VALUES ('D2117', 'MWF', 12, 'cis 162');
SQL> INSERT INTO classes VALUES ('C100', 'MF', 11, 'mth 101');
SQL> INSERT INTO classes VALUES ('D2117', 'MF', 11, 'mth 102');
SQL> INSERT INTO classes VALUES ('F2120', 'TuTh', 10, 'art 150');
SQL> INSERT INTO classes VALUES ('F2120', 'MWF', 10, 'art 150');
SQL> INSERT INTO classes VALUES ('A1200', 'MWF', 14, 'cis 361');
SQL> INSERT INTO classes VALUES ('A1200', 'TuTh', 15, 'cis 353');
SQL>
SQL> --enrolled_in(student ssn, class room num, class time)
SQL> INSERT INTO enrolled_in VALUES ('100000001', 'D2117', 'MWF');
SQL> INSERT INTO enrolled_in VALUES ('100000001', 'A117', 'MWF');
SQL> INSERT INTO enrolled_in VALUES ('100000001', 'C100', 'MF');
SQL> INSERT INTO enrolled_in VALUES ('100000004', 'F2120', 'TuTh');
SQL> INSERT INTO enrolled_in VALUES ('100000004', 'A117', 'TuTh');
SQL> INSERT INTO enrolled_in VALUES ('100000005', 'A1200', 'TuTh');
SQL> INSERT INTO enrolled_in VALUES ('100000005', 'A1200', 'MWF');
SQL> INSERT INTO enrolled_in VALUES ('100000005', 'D2117', 'MWF');
SQL> INSERT INTO enrolled_in VALUES ('100000006', 'C100', 'TuTh');
SQL> INSERT INTO enrolled_in VALUES ('100000006', 'D2117', 'MF');
SQL> INSERT INTO enrolled_in VALUES ('100000006', 'F2120', 'MWF');
SQL>
SQL> SET FEEDBACK ON
SQL> COMMIT;

```

Commit complete.

```

SQL>
SQL> --
SQL> -- PRINTING OUT THE DATABASE
SQL> --
SQL>
SQL> SELECT * FROM student;

```

STUDENT_S	STUDENT_NAME	STUDENT_BD
100000001	Drew VM	1989-08-12
	computer science	60
100000002	Andy Hung	1996-08-19
	computer science	40
100000003	Joey Coscarelli	1992-12-08
	computer science	50

STUDENT_S	STUDENT_NAME	STUDENT_BD
100000004	Morgan Freeman	1963-02-10
	theatre	10
100000005	Judi Dench	1972-11-19

theatre	26
100000006 Philip Philips	1990-12-27
mechanical engineering	31

6 rows selected.

SQL> SELECT * FROM classes;

CLASS	CLASS_TIME	CL	CLASS_COURSE
A117	MWF	13	phy 220
A117	TuTh	13	phy 220
C100	TuTh	13	mth 325
D2117	MWF	12	cis 162
C100	MF	11	mth 101
D2117	MF	11	mth 102
F2120	TuTh	10	art 150
F2120	MWF	10	art 150
A1200	MWF	14	cis 361
A1200	TuTh	15	cis 353

10 rows selected.

SQL> SELECT * FROM teacher;

TE	TEACHER_NAME
13	Dave Wilson
12	Queen Takerook
11	Jane Doe
10	Huey Lewis
14	John Smith
15	Peter Parker

6 rows selected.

SQL> SELECT * FROM club;

CLUB_SUBJECT	CLUB_MEET	CL
physics	tuesday	13
chess	thursday	12
math	tuesday	11
news	monday	10

4 rows selected.

SQL> SELECT * FROM room;

ROOM_	ROOM_LOCATION	ROOM_MAX_OCCUPANCY	ROOM_PROJECTORS
A117	mac	35	1
D2117	mac	40	2
C100	pad	12	0
F2120	pac	32	0
A1200	man	36	2

5 rows selected.

SQL> SELECT * FROM member_of;

MEMBER_ST	MEMBER_CLUB_SUB	MEMBER_DAT
100000001	physics	2016-01-01
100000001	math	2016-02-01

```

100000003 chess          2015-02-14
100000004 chess          2016-10-22
100000004 news           2016-01-13
100000005 news           2015-10-12
100000005 math           2015-05-13
100000005 chess          2014-05-13
100000005 physics        2014-05-13
100000006 math           2016-12-21

```

10 rows selected.

```
SQL> SELECT * FROM enrolled_in;
```

```

ENROLLED_  ENROL  ENROLLED_CLASS_TIME
-----
100000001 D2117 MWF
100000001 A117  MWF
100000001 C100  MF
100000004 F2120 TuTh
100000004 A117  TuTh
100000005 A1200 TuTh
100000005 A1200 MWF
100000005 D2117 MWF
100000006 C100  TuTh
100000006 D2117 MF
100000006 F2120 MWF

```

11 rows selected.

```
SQL> SELECT * FROM student_phone;
```

```

PHONE_STU  PHONE_NUMB
-----
100000001 364-1102
100000001 364-8901
100000003 867-5309
100000005 312-2150
100000005 312-7775
100000006 616-1234

```

6 rows selected.

```

SQL>
SQL> --
SQL> -- SQL QUERIES
SQL> --
SQL>
SQL> -- Q1 - self join
SQL> -- Get pairs of students who are in the same major
SQL> SELECT S1.student_name, S2.student_name
2  FROM student S1, student S2
3  WHERE S1.student_major = S2.student_major AND
4         S1.student_ssn < S2.student_ssn;

```

```

STUDENT_NAME          STUDENT_NAME
-----
Drew VM               Andy Hung
Andy Hung             Joey Coscarelli
Drew VM               Joey Coscarelli
Morgan Freeman        Judi Dench

```

4 rows selected.

```

SQL>
SQL> -- Q2 - group by, having, order by
SQL> -- Find every student who is enrolled in more than 2 classes
SQL> SELECT S.student_name, S.student_ssn, COUNT(*)

```

```

2 FROM student S, enrolled_in E
3 WHERE S.student_ssn = E.enrolled_student_ssn
4 GROUP BY S.student_name, S.student_ssn
5 HAVING COUNT(*) > 2
6 ORDER BY S.student_name;

```

STUDENT_NAME	STUDENT_S	COUNT(*)
Drew VM	100000001	3
Judi Dench	100000005	3
Philip Philips	100000006	3

3 rows selected.

```

SQL>
SQL> -- Q3 - joining of 4 tables
SQL> -- Find the name and course name of every student enrolled in a class
taught by teachers named Huey
SQL> SELECT S.student_name, C.class_course
2 FROM student S, enrolled_in E, teacher T, classes C
3 WHERE T.teacher_name like '%Huey%' AND
4 T.teacher_id = C.class_tid AND
5 C.class_room_num = E.enrolled_class_room_num AND
6 C.class_time = E.enrolled_class_time AND
7 E.enrolled_student_ssn = S.student_ssn;

```

STUDENT_NAME	CLASS_COURSE
Morgan Freeman	art 150
Philip Philips	art 150

2 rows selected.

```

SQL>
SQL> -- Q4 - division and minus
SQL> -- Find every student who is a member of every club
SQL> SELECT S.student_name
2 FROM student S
3 WHERE NOT EXISTS ((SELECT C.club_subject
4 FROM club C)
5 MINUS
6 (SELECT C.club_subject
7 FROM club C, member_of M
8 WHERE S.student_ssn = M.member_student_ssn AND
9 M.member_club_subject = C.club_subject));

```

STUDENT_NAME
Judi Dench

1 row selected.

```

SQL>
SQL> -- Q5 - Maximum
SQL> -- Find the max credit hours of students who are majoring in theatre
SQL> SELECT MAX(student_credit_hours)
2 FROM student S
3 WHERE S.student_major = 'theatre';

```

MAX(STUDENT_CREDIT_HOURS)
26

26

1 row selected.

```

SQL>
SQL> -- Q6 - Non correlated

```



```
SQL> -- Find every student who is not enrolled in any class
SQL> SELECT S.student_name
  2 FROM student S
  3 WHERE S.student_ssn NOT IN (Select E.enrolled_student_ssn
  4                               FROM enrolled_in E);
```

STUDENT_NAME

 Joey Coscarelli
 Andy Hung

2 rows selected.

```
SQL>
SQL> -- Q7 - correlated
SQL> -- Find every teacher who is not running any clubs
SQL> SELECT T.teacher_name
  2 FROM teacher T
  3 WHERE NOT EXISTS          (SELECT *
  4                               FROM club C
  5                               WHERE C.club_tid = T.teacher_id);
```

TEACHER_NAME

 John Smith
 Peter Parker

2 rows selected.

```
SQL>
SQL> -- Q8 - Outer Join
SQL> -- Find the name, ssn, and class times for every student. Also show if a
student has no class times
SQL> SELECT S.student_name, S.student_ssn, E.enrolled_class_time
  2 FROM student S LEFT OUTER JOIN enrolled_in E ON E.enrolled_student_ssn =
S.student_ssn;
```

STUDENT_NAME	STUDENT_S	ENROLLED_CLASS_TIME
-----	-----	-----
Drew VM	100000001	MWF
Drew VM	100000001	MF
Drew VM	100000001	MWF
Morgan Freeman	100000004	TuTh
Morgan Freeman	100000004	TuTh
Judi Dench	100000005	MWF
Judi Dench	100000005	TuTh
Judi Dench	100000005	MWF
Philip Philips	100000006	TuTh
Philip Philips	100000006	MF
Philip Philips	100000006	MWF

STUDENT_NAME	STUDENT_S	ENROLLED_CLASS_TIME
-----	-----	-----
Joey Coscarelli	100000003	
Andy Hung	100000002	

13 rows selected.

```
SQL>
SQL> -- Q9 - RANK
SQL> -- Find what rank is associated with 40 credit hours
SQL> SELECT RANK (40) WITHIN GROUP
  2 (ORDER BY student_credit_hours) "Rank of credit hours 40"
  3 FROM student;
```

Rank of credit hours 40

1 row selected.

```
SQL>
SQL> -- Q10 - Top N
SQL> -- Of all students, which 3 have the least credit hours
SQL> SELECT student_name
      2 FROM (SELECT * FROM student ORDER BY student_credit_hours)
      3 WHERE ROWNUM < 4;
```

STUDENT_NAME

```
-----
Morgan Freeman
Judi Dench
Philip Philips
```

3 rows selected.

```
SQL>
SQL> --
SQL> -- TESTS FOR INTEGRITY CONSTRAINTS
SQL> --
SQL>
SQL> -- Testing: <teacher_PK>
SQL> UPDATE teacher SET teacher_id = 13 WHERE teacher_id = 15;
UPDATE teacher SET teacher_id = 13 WHERE teacher_id = 15
*
ERROR at line 1:
ORA-00001: unique constraint (MURTHUMB.TEACHER_PK) violated
```

```
SQL>
SQL> -- Testing: <student_phone_FK_1>
SQL> UPDATE student_phone SET phone_student_ssn = 100000009 WHERE
phone_student_ssn = 100000003;
```

1 row updated.

```
SQL>
SQL> -- Testing: <club_IC_1>
SQL> INSERT INTO club VALUES ('new club', 'Invalid', 15);
INSERT INTO club VALUES ('new club', 'Invalid', 15)
*
ERROR at line 1:
ORA-02290: check constraint (MURTHUMB.CLUB_IC_1) violated
```

```
SQL>
SQL> -- Testing: <room_IC_1>
SQL> INSERT INTO room VALUES ('New1', 'mac', 20, 3);
INSERT INTO room VALUES ('New1', 'mac', 20, 3)
*
ERROR at line 1:
ORA-02290: check constraint (MURTHUMB.ROOM_IC_1) violated
```

```
SQL>
SQL> COMMIT;
COMMIT
*
ERROR at line 1:
ORA-02091: transaction rolled back
ORA-02291: integrity constraint (MURTHUMB.STUDENT_PHONE_FK_1) violated - parent
key not found
```

```
SQL> --  
SQL> SPOOL OFF
```