**SQL Lab**

Mingjian Shi

1. Define the schema of the LIPSCOMB\_Student\_Registration database

CREATE DATABASE Lipscomb\_Student\_Registration;

a. Defining the Database

Define a database schema by constructing SQL CREATE TABLE commands for each of the following relations (7 tables):

1. Primary keys i – vi were chosen as they are unique identifiers for each of the specific table. Primary key g was created from an auto increment, as there is not one unique key.
   * 1. Student: S\_ID
     2. Faculty: F\_ID
     3. Course: Course\_ID
     4. Course\_Section: C\_Sec\_ID
     5. Location: LOC\_ID
     6. Term: Term\_ID
     7. Enrollment: enr\_ID

Foreign keys:

1. Enrollment
   1. S\_ID – cascade updates, no action on delete
   2. C\_Sec\_ID - cascade updates, restrict delete
2. Course\_Section
   1. Course\_ID – cascade updates, no action on delete
   2. Term\_ID – cascade updates, no action on delete
   3. F\_ID – cascade updates, no action on delete
   4. LOC\_ID - cascade updates, no action on delete
3. Student
   1. F\_ID - cascade updates, restrict delete
4. Faculty
   1. LOD\_ID - cascade updates, restrict delete

2.1 CREATE TABLE `STUDENT` (

`S\_ID` INTEGER PRIMARY KEY AUTO\_INCREMENT,

`S\_LAST` VARCHAR(50) NOT NULL,

`S\_FIRST` VARCHAR(50) NOT NULL,

`S\_MI` VARCHAR(1) NOT NULL,

`S\_ADDRESS` VARCHAR(75) NOT NULL,

`S\_CITY` VARCHAR(50) NOT NULL,

`S\_STATE` VARCHAR(2) NOT NULL,

`S\_ZIP` VARCHAR(5) NOT NULL,

`S\_PHONE` VARCHAR(15) NOT NULL,

`S\_CLASS` VARCHAR(5) NOT NULL,

S\_DOB DATE NOT NULL,

`S\_PIN` VARCHAR(4) NOT NULL,

`DATE\_ENROLLED` DATE,

`F\_ID` INTEGER,

FOREIGN KEY (`F\_ID`)

REFERENCES FACULTY (`F\_ID`)

ON UPDATE CASCADE

ON DELETE RESTRICT

);

2.2 CREATE TABLE `FACULTY` (

`F\_ID` INTEGER PRIMARY KEY AUTO\_INCREMENT,

`F\_LAST` VARCHAR(50) NOT NULL,

`F\_FIRST` VARCHAR(50) NOT NULL,

`F\_MI` VARCHAR(1),

`F\_PHONE` VARCHAR(15),

`F\_RANK` VARCHAR(30) NOT NULL,

`F\_SUPER` INTEGER,

`F\_PIN` VARCHAR(10) NOT NULL,

`LOC\_ID` INTEGER,

FOREIGN KEY (`LOC\_ID`)

REFERENCES location (`LOC\_ID`)

ON UPDATE CASCADE

ON DELETE RESTRICT

);

2.3 CREATE TABLE `COURSE` (

`COURSE\_ID` INTEGER PRIMARY KEY AUTO\_INCREMENT,

`COURSE\_NO` VARCHAR(10) NOT NULL,

`COURSE\_NAME` VARCHAR(50) NOT NULL,

`CREDITS` TINYINT NOT NULL

);

2.4 CREATE TABLE `COURSE\_SECTION` (

`C\_SEC\_ID` INTEGER PRIMARY KEY AUTO\_INCREMENT,

`COURSE\_ID` INTEGER NOT NULL,

`TERM\_ID` INTEGER NOT NULL,

`SEC\_NUM` INTEGER NOT NULL,

`F\_ID` INTEGER NOT NULL,

`MTG\_DAYS` VARCHAR(7) NOT NULL,

`START\_TIME` TIME NOT NULL,

`END\_TIME` TIME NOT NULL,

`LOC\_ID` INTEGER,

`MAX\_ENRL` INTEGER,

FOREIGN KEY (`COURSE\_ID`)

REFERENCES course (`COURSE\_ID`)

ON UPDATE CASCADE

ON DELETE NO ACTION,

FOREIGN KEY (`TERM\_ID`)

REFERENCES term (`TERM\_ID`)

ON UPDATE CASCADE

ON DELETE NO ACTION,

FOREIGN KEY (`F\_ID`)

REFERENCES faculty (`F\_ID`)

ON UPDATE CASCADE

ON DELETE NO ACTION,

FOREIGN KEY (`LOC\_ID`)

REFERENCES location (`LOC\_ID`)

ON UPDATE CASCADE

ON DELETE NO ACTION

);

2.5 CREATE TABLE `ENROLLMENT` (

`ENR\_ID` INTEGER PRIMARY KEY AUTO\_INCREMENT,

`S\_ID` INTEGER NOT NULL,

`C\_SEC\_ID` INTEGER NOT NULL,

`GRADE` VARCHAR(5) DEFAULT NULL,

FOREIGN KEY (`S\_ID`)

REFERENCES STUDENT (`S\_ID`)

ON UPDATE CASCADE

ON DELETE NO ACTION,

FOREIGN KEY (`C\_SEC\_ID`)

REFERENCES COURSE\_SECTION (`C\_SEC\_ID`)

ON UPDATE CASCADE

ON DELETE RESTRICT

);

2.6 CREATE TABLE `LOCATION` (

`LOC\_ID` INTEGER PRIMARY KEY AUTO\_INCREMENT,

`BLDG\_CODE` VARCHAR(10) NOT NULL,

`ROOM` VARCHAR(5) NOT NULL,

`CAPACITY` INTEGER NOT NULL

);

2.7 CREATE TABLE `TERM` (

`TERM\_ID` INTEGER PRIMARY KEY AUTO\_INCREMENT,

`TERM\_DESC` VARCHAR(20) NOT NULL,

`STATUS` VARCHAR(6) NOT NULL,

`START\_DATE` DATE NOT NULL

);

Question2

1. Copy the data .doc file and paste them to the Microsoft excel and save them as seven .CSV file. Now the data were transformed into comma-separated files.
2. Steps to do the to do the conversion of the comma-separated files.
   1. From Oracle SQL application express Utilites select Data Workshop

Data Load

* [Text Data](javascript:apex.navigation.dialog('f?p=4300:230:112048269355573::NO:18,19,21,22,24,25:F4300_P230_LOAD_FROM:UPLOAD\u0026p_dialog_cs=E4Nxc2wWFgmjleZHQ-re-rnmGmb4xgj-gQYuPag3F5NFGoRMiDAkIPIi432cIxTEBdi_M9p2O0-cw14sROo-yg',%7btitle:'Load%20Data%20-%20Target%20and%20Method',height:'500',width:'800',maxWidth:'1200',modal:true,dialog:null,resizable:true,minWidth:600,minHeight:500%7d,'a-Dialog--wizard',apex.jQuery('#R170288209634031042'));)
* [XML Data](javascript:apex.navigation.dialog('f?p=4300:14:112048269355573::NO:14::\u0026p_dialog_cs=NkYdJfRnauO_osw0YJLwgelt4iozh3vTvzStjfz7mCu2-ywB6xWDcNnqPCy8cAeOX3bbjinwoKSqJeJl64HHsA',%7btitle:'Load%20XML%20Data',height:'480',width:'800',maxWidth:'1200',modal:true,dialog:null,resizable:true,minWidth:500,minHeight:400%7d,'a-Dialog--wizard',apex.jQuery('#R170288209634031042'));)
* [Spreadsheet Data](javascript:apex.navigation.dialog('f?p=4300:230:112048269355573::NO:200,210,220,230,240,260,270::\u0026p_dialog_cs=E4Nxc2wWFgmjleZHQ-re-rnmGmb4xgj-gQYuPag3F5NFGoRMiDAkIPIi432cIxTEBdi_M9p2O0-cw14sROo-yg',%7btitle:'Load%20Data%20-%20Target%20and%20Method',height:'500',width:'800',maxWidth:'1200',modal:true,dialog:null,resizable:true,minWidth:600,minHeight:500%7d,'a-Dialog--wizard',apex.jQuery('#R170288209634031042'));)

Data Unload

* [to Text](javascript:apex.navigation.dialog('f?p=4300:150:112048269355573::NO:150,180::\u0026p_dialog_cs=b4ZJbi8lmLL-iZg-wO_xpSMHBLRit4H8qkAahEze65n-3r0ijZp2pOanvFX20g-wO5OPTJ3Zc5D6kBqHl-6R9g',%7btitle:'Unload%20to%20Text%20-%20Columns',height:'480',width:'800',maxWidth:'1200',modal:true,dialog:null,resizable:true,minWidth:500,minHeight:400%7d,'a-Dialog--wizard',apex.jQuery('#R170288516691042569'));)
* [to XML](javascript:apex.navigation.dialog('f?p=4300:90:112048269355573::NO:90::\u0026p_dialog_cs=c_g-g9RKKnU0IDEFepIIMnKXuOU4_yrhnikaSVay-XvhDAQmk5Qo_AKkt28wepoi7TPK_ybXj-16L-nJArWAiQ',%7btitle:'Unload%20to%20XML%20-%20Columns',height:'480',width:'800',maxWidth:'1200',modal:true,dialog:null,resizable:true,minWidth:500,minHeight:400%7d,'a-Dialog--wizard',apex.jQuery('#R170288516691042569'));)

Repository

* [Import Repository](https://apex.oracle.com/pls/apex/f?p=4300:8:112048269355573::NO:RP::)
* [Spreadsheet Imports](https://apex.oracle.com/pls/apex/f?p=4300:11:112048269355573::NO:RP::)

Use Data load function to do text data loading

Load To:

Load To:Existing table  
New table

Load From:

Load From:Upload file (comma separated or tab delimited)  
Copy and paste

Then copy the text from opened .CSV file and paste to the

Copy the data you want to import from a spreadsheet program, such as Microsoft Excel, and paste it into the Data field.

Data(Value Required)



Separator(Value Required)



Enclosed By



First row contains column names.

[**[https://apex.oracle.com/i/f_spacer.gif](javascript:void(0);)Globalization**](javascript:void(0);)

Save the context to the table.

Question 3

* 1. INSERT INTO course\_section (C\_SEC\_ID,COURSE\_ID,TERM\_ID,SEC\_NUM,F\_ID,MTG\_DAYS,START\_TIME,END\_TIME,LOC\_ID,MAX\_ENRL) VULUES (12, 2, 6, 2, 2, “MTWRF”, 10:00 AM, 11:30 AM, 5, 35);

**ORA-00926: missing VALUES keyword**

INSERT INTO course\_section (C\_SEC\_ID,COURSE\_ID,TERM\_ID,SEC\_NUM,F\_ID,MTG\_DAYS,START\_TIME,END\_TIME,LOC\_ID,MAX\_ENRL) VULUES (12, 2, 6, 2, 2, “MTWRF”, 9:00 AM, 10:30 AM, 6, 35);

**ORA-00926: missing VALUES keyword**

INSERT INTO course\_section (C\_SEC\_ID,COURSE\_ID,TERM\_ID,SEC\_NUM,F\_ID,MTG\_DAYS,START\_TIME,END\_TIME,LOC\_ID,MAX\_ENRL) VULUES (2, 1, 4, 2, 3, “TR”, 9:30 AM, 10:45 AM, 4, 35);

**ORA-00926: missing VALUES keyword**

* 1. INSERT INTO Faculty (F\_ID,F\_LAST,F\_FIRST,F\_MI,LOC\_ID,F\_PHONE,F\_RANK,F\_SUPER,F\_PIN) VALUES (4, “Brown”, “Colin”, “D”, 11, “3253456789”, “Assistant”, 4, 9871);

**ORA-00911: invalid character**

INSERT INTO Faculty (F\_ID,F\_LAST,F\_FIRST,F\_MI,LOC\_ID,F\_PHONE,F\_RANK,F\_SUPER,F\_PIN) VALUES (6, “Reeves”, “Bob”, “S”, 15, “3256789012”, “Full”, , 1234);

**ORA-00911: invalid character**

INSERT INTO Faculty (F\_ID,F\_LAST,F\_FIRST,F\_MI,LOC\_ID,F\_PHONE,F\_RANK,F\_SUPER,F\_PIN) VALUES (6, “Reeves”, “Bob”, “S”, 10, “3256789012”, “Assistant”, 7, 1234);

**ORA-00911: invalid character**

INSERT INTO Faculty (F\_ID,F\_LAST,F\_FIRST,F\_MI,LOC\_ID,F\_PHONE,F\_RANK,F\_SUPER,F\_PIN) VALUES (6, “Reeves”, “Bob”, “S”, 10, “3255678901”, “Assistant”, 2, 1234);

**ORA-00911: invalid character**

* 1. INSERT INTO course (COURSE\_ID, COURSE\_NO,COURSE\_NAME,CREDITS) VALUES (4, “CS 120”, “Intro. to Programming in C++”, 3);

**ORA-00911: invalid character**

* 1. DELETE FROM Location

WHERE LOC\_ID=11;

1 row(s) deleted.

0.02 seconds

* 1. DELETE FROM TERM

WHERE TERM\_ID=4;

1 row(s) deleted.

0.01 seconds

Question 4

* 1. SELECT STUDENT.S\_ID, S\_LAST, S\_FIRST, ENROLLMENT.GRADE from ENROLLMENT inner JOIN STUDENT on ENROLLENT.S\_ID= STUDENT.S\_ID WHERE ENROLLMENT.GRADE in (‘A’, ‘B’);
  2. SELECT TERM\_DESC from TERM

Where TERM\_DESC like '%2007%';

|  |  |  |
| --- | --- | --- |
| **TERM\_DESC** | | |
| Spring 2007 | | |
| Summer 2007 | | |
| 2 rows returned in 0.02 seconds |  | [Download](https://apex.oracle.com/pls/apex/f?p=4500:1204:1162322317881::::P1204_BROWSER_LANG:en) |

c. SELECT Loc\_ID, Bldg\_Code, Room, Capacity FROM LOCATION

order by Bldg\_Code ASC, ROOM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LOC\_ID** | **BLDG\_CODE** | **ROOM** | | **CAPACITY** |
| 5 | BUS | 105 | | 42 |
| 8 | BUS | 211 | | 55 |
| 10 | BUS | 402 | | 1 |
| 6 | BUS | 404 | | 35 |
| 7 | BUS | 421 | | 35 |
| 9 | BUS | 424 | | 1 |
| 1 | CR | 101 | | 150 |
| 3 | CR | 103 | | 35 |
| 4 | CR | 105 | | 35 |
| 2 | CR | 202 | | 40 |
| More than 10 rows available. Increase rows selector to view more rows. | | | | |
| 10 rows returned in 0.00 seconds | |  | [Download](https://apex.oracle.com/pls/apex/f?p=4500:1204:337033404308::::P1204_BROWSER_LANG:en) |

1. SELECT COURSE\_NO, COURSE\_NAME,

SUM(CREDITS \* 730) AS TUITION\_CHARGE

FROM COURSE;

e.

f. SELECT COUNT(DISTINCT E.GRADE) COURSE\_COUNT

FROM ENROLLMENT E

JOIN STUDENT S

ON E.S\_ID = S.S\_ID

WHERE S\_FIRST = 'Lisa'

AND S\_LAST = 'Johnson'

AND (GRADE IS NOT NULL);

|  |
| --- |
| **GRADE** |
| B |

g. SELECT BLDG\_CODE, SUM(CAPACITY)

FROM LOCATION

GROUP BY BLDG\_CODE

HAVING SUM(CAPACITY) > 100;

|  |  |
| --- | --- |
| **BLDG\_CODE** | **SUM(CAPACITY)** |
| CR | 260 |
| BUS | 170 |

h. SELECT S.S\_ID, S.S\_LAST, S.S\_FIRST, S.F\_ID, F.F\_LAST

FROM STUDENT S

JOIN FACULTY F ON S.F\_ID = F.F\_ID;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S\_ID** | **S\_LAST** | **S\_FIRST** | **F\_ID** | **F\_LAST** |
| 1 | Jones | Tammy | 1 | Marx |
| 2 | Perez | Jorge | 1 | Marx |
| 3 | Marsh | John | 1 | Marx |
| 4 | Smith | Mike | 2 | Zhulin |
| 6 | Nguyen | Ni | 3 | Langley |
| 5 | Johnson | Lisa | 4 | Brown |

1. SELECT F.F\_LAST

FROM FACULTY F

JOIN COURSE\_SECTION CS

ON CS.F\_ID = F.F\_ID

JOIN TERM T

ON CS.TERM\_ID = T.TERM\_ID

WHERE TERM\_DESC = 'Summer 2008';

|  |
| --- |
| **F\_LAST** |
| Marx |
| Zhulin |
| Langley |
| 3 rows returned in 0.03 seconds |

1. SELECT COURSE\_NAME,GRADE, S.S\_ID

FROM ENROLLMENT E

JOIN STUDENT S

ON E.S\_ID = S.S\_ID

JOIN COURSE\_SECTION CS

ON E.C\_SEC\_ID = CS.C\_SEC\_ID

JOIN COURSE C

ON CS.COURSE\_ID = C.COURSE\_ID

WHERE S\_FIRST = 'Tammy'

AND S\_LAST = 'Jones';

|  |  |  |
| --- | --- | --- |
| **COURSE\_NAME** | **GRADE** | **S\_ID** |
| Intro. to Info. Tech. | A | 1 |

1. SELECT S\_LAST, S\_FIRST, S\_PHONE

FROM STUDENT S

UNION

SELECT F\_LAST, F\_FIRST, F\_PHONE

FROM FACULTY F

|  |  |  |
| --- | --- | --- |
| **S\_LAST** | **S\_FIRST** | **S\_PHONE** |
| Brown | Jonnel | 3254567890 |
| Johnson | Lisa | 3255432109 |
| Jones | Tammy | 3250987654 |
| Langley | Colin | 3253456789 |
| Marsh | John | 3257654321 |
| Marx | Teresa | 3251234567 |
| Nguyen | Ni | 3254321098 |
| Perez | Jorge | 3258765432 |
| Sealy | James | 3255678901 |
| Smith | Mike | 3256543210 |
| More than 10 rows available. Increase rows selector to view more rows. | | |

Question 5

1. SELECT S\_FIRST, S\_LAST FROM STUDENT S

WHERE S\_CLASS = ( SELECT S\_CLASS FROM STUDENT S WHERE S\_FIRST = 'Jorge' AND S\_LAST = 'Perez');

|  |  |  |
| --- | --- | --- |
| **S\_FIRST** | | **S\_LAST** |
| Tammy | | Jones |
| Jorge | | Perez |
| 2 rows returned in 0.02 seconds | |  | [Download](https://apex.oracle.com/pls/apex/f?p=4500:1204:103322682671634::::P1204_BROWSER_LANG:en) |

1. SELECT DISTINCT S\_LAST, S\_FIRST

FROM STUDENT S

JOIN ENROLLMENT E

ON S.S\_ID = E.S\_ID

WHERE E.C\_SEC\_ID IN (

SELECT E.C\_SEC\_ID

FROM STUDENT S

JOIN ENROLLMENT E

ON S.S\_ID = E.S\_ID

WHERE S\_FIRST = 'Jorge'

AND S\_LAST = 'Perez');

|  |  |
| --- | --- |
| **S\_LAST** | **S\_FIRST** |
| Johnson | Lisa |
| Jones | Tammy |
| Perez | Jorge |
| Marsh | John |

1. SELECT DISTINCT S\_LAST, S\_FIRST

FROM STUDENT S

JOIN ENROLLMENT E

ON S.S\_ID = E.S\_ID

WHERE E.C\_SEC\_ID IN (

SELECT E.C\_SEC\_ID

FROM STUDENT S

JOIN ENROLLMENT E

ON S.S\_ID = E.S\_ID

WHERE S\_FIRST = 'Jorge'

AND S\_LAST = 'Perez'

)

AND S\_CLASS = (

SELECT S\_CLASS

FROM STUDENT S

WHERE S\_FIRST = 'Jorge'

AND S\_LAST = 'Perez'

);

|  |  |
| --- | --- |
| **S\_LAST** | **S\_FIRST** |
| Jones | Tammy |
| Perez | Jorge |

1. SELECT S.S\_FIRST, S.S\_LAST

FROM STUDENT S

JOIN ENROLLMENT E

ON S.S\_ID = E.S\_ID

JOIN COURSE\_SECTION CS

ON E.C\_SEC\_ID = CS.C\_SEC\_ID

WHERE CS.C\_SEC\_ID IN (

SELECT CS.C\_SEC\_ID

FROM STUDENT S

JOIN ENROLLMENT E

ON S.S\_ID = E.S\_ID

JOIN COURSE\_SECTION CS

ON E.C\_SEC\_ID = CS.C\_SEC\_ID

JOIN LOCATION L

ON CS.LOC\_ID = L.LOC\_ID

WHERE S.S\_ID IN (

SELECT S.S\_ID

FROM STUDENT S

WHERE S.S\_FIRST = 'Jorge'

and s.S\_LAST = 'Perez')

AND BLDG\_CODE = 'CR');

|  |  |
| --- | --- |
| **S\_FIRST** | **S\_LAST** |
| Lisa | Johnson |
| John | Marsh |
| Jorge | Perez |
| Tammy | Jones |

1. SELECT COURSE\_NAME

FROM COURSE C

JOIN COURSE\_SECTION CS

ON C.COURSE\_ID = CS.COURSE\_ID

JOIN ENROLLMENT E

ON CS.C\_SEC\_ID = E.C\_SEC\_ID

JOIN STUDENT S

ON E.S\_ID = S.S\_ID

WHERE S\_CLASS <> 'SR'

UNION

SELECT COURSE\_NAME

FROM COURSE C

JOIN COURSE\_SECTION CS

ON C.COURSE\_ID = CS.COURSE\_ID

JOIN ENROLLMENT E

ON CS.C\_SEC\_ID = E.C\_SEC\_ID

JOIN STUDENT S

ON E.S\_ID = S.S\_ID

WHERE TERM\_ID = 6;

|  |
| --- |
| **COURSE\_NAME** |
| Intro. to Info. Tech. |
| Intro. to Info. Tech. |

1. SELECT DISTINCT COURSE\_NAME FROM COURSE C

JOIN COURSE\_SECTION CS

ON C.COURSE\_ID = CS.COURSE\_ID

JOIN ENROLLMENT E

ON CS.C\_SEC\_ID = E.C\_SEC\_ID

JOIN STUDENT S

ON E.S\_ID = S.S\_ID

WHERE S\_CLASS <> 'SR'

AND COURSE\_NAME IN (

SELECT COURSE\_NAME

FROM COURSE C

JOIN COURSE\_SECTION CS

ON C.COURSE\_ID = CS.COURSE\_ID

JOIN ENROLLMENT E

ON CS.C\_SEC\_ID = E.C\_SEC\_ID

JOIN STUDENT S

ON E.S\_ID = S.S\_ID

WHERE TERM\_ID = 6);

|  |
| --- |
| **COURSE\_NAME** |
| Intro. to Info. Tech. |

1. SELECT DISTINCT COURSE\_NAME FROM COURSE C

JOIN COURSE\_SECTION CS

ON C.COURSE\_ID = CS.COURSE\_ID

JOIN ENROLLMENT E

ON CS.C\_SEC\_ID = E.C\_SEC\_ID

JOIN STUDENT S

ON E.S\_ID = S.S\_ID

WHERE S\_CLASS IN ('FR','SO','JR')

AND COURSE\_NAME NOT IN (

SELECT COURSE\_NAME FROM COURSE C

JOIN COURSE\_SECTION CS

ON C.COURSE\_ID = CS.COURSE\_ID

JOIN ENROLLMENT E

ON CS.C\_SEC\_ID = E.C\_SEC\_ID

JOIN STUDENT S

ON E.S\_ID = S.S\_ID

WHERE TERM\_ID = 6);

no data found

1. Is the F\_Super number the same withF\_ID?

Question 6

REPLACE INTO LOCATION (L.Room) VALUE(BUS211)

WHERE L.LOCATION=(SELECT L.LOCATION, F.F\_LAST FROM FACULTY F

JOIN LOCATION L ON

L.LOC\_ID=F.LOC\_ID

WHERE F.F\_LAST=”Brown” );

CREATE TABLE ENROLLMENT\_NUMBERS (

COURSE\_ID INTEGER,

C\_SEC\_ID INTEGER,

ENROLLMENTS INTEGER);

INSERT INTO ENROLLMENT\_NUMBERS

SELECT COURSE\_ID, CS.C\_SEC\_ID, COUNT(S\_ID) AS ENROLLMENTS

FROM COURSE\_SECTION CS

JOIN ENROLLMENT E

ON CS.C\_SEC\_ID = E.C\_SEC\_ID

JOIN TERM T

ON CS.TERM\_ID = T.TERM\_ID

WHERE T.TERM\_DESC = 'SPRING 2008'

GROUP BY COURSE\_ID, CS.C\_SEC\_ID;

SELECT \* FROM ENROLLMENT\_NUMBERS;