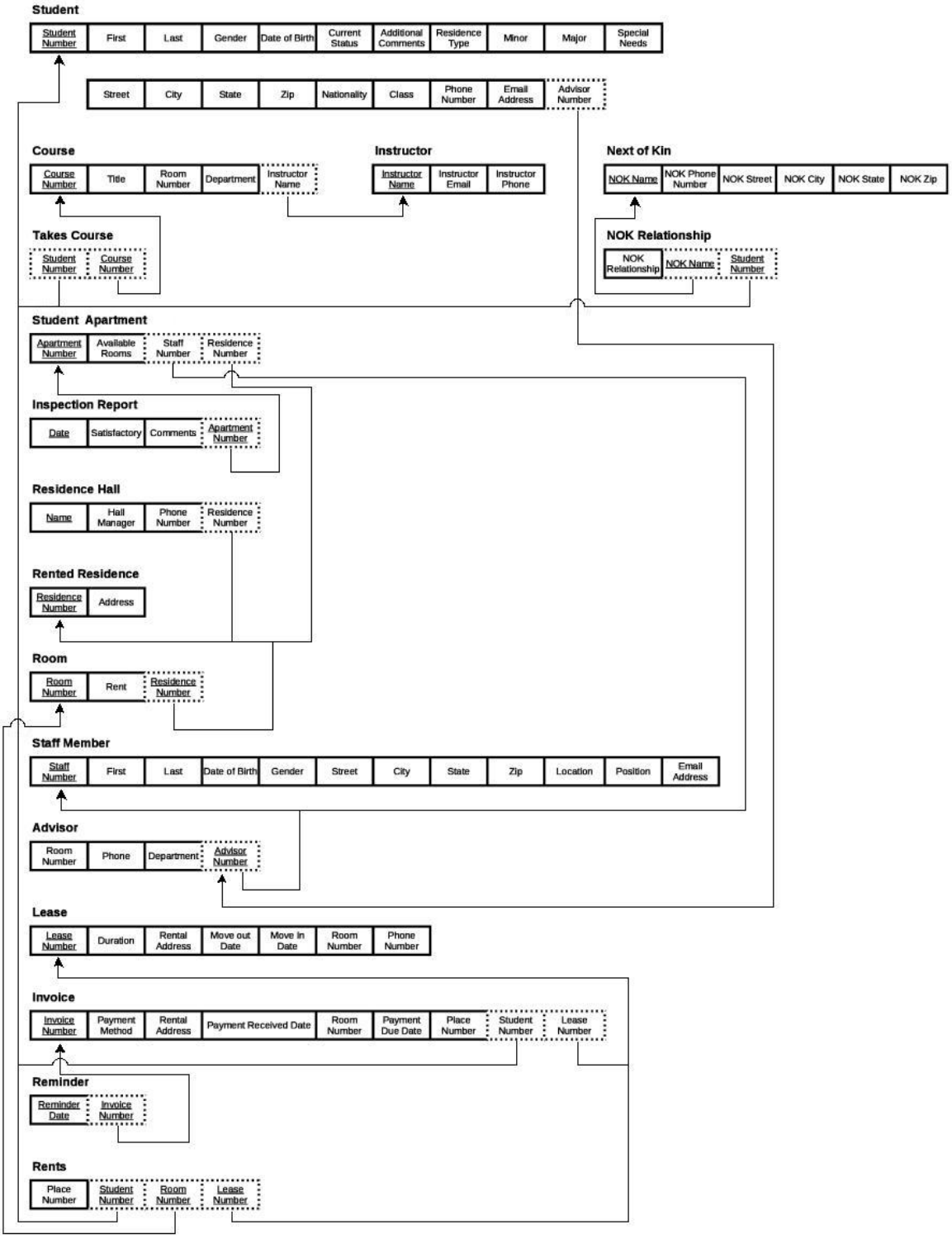
Normalized Relational Model Schema



Lease

Team Member Responsible: Enrique Carbajal

Table Comments

TABLE_NAME	TABLE_TYPE	COMMENTS
LEASE	TABLE	A lease is a paper stating the terms that bind a landlord and tenant.

Column Comments

TABLE_NAME	COLUMN_NAME	COMMENTS
LEASE	LEASE_NUMBER	The unique number associated with the lease -- Format:10000
	DURATION	The time period of the lease -- Format:1
	RENTAL_ADDRESS	The address of the rental property-- Format:4135 Sierra St. San Bernardino, CA. 92408
	MOVE_OUT_DATE	The date the renters have to vacate the property -- Format:12/15/2012
	MOVE_IN_DATE	The date that renters take possession -- Format:8/30/2012
	ROOM_NUMBER	Which room is being leased -- Format: 3001
	PHONE_NUMBER	The phone number of the lessee-- Format: 909-303-3030

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Enrique Carbajal
*/

CREATE TABLE Lease (
Lease_Number VARCHAR2(5)
     CONSTRAINT Lease_PK PRIMARY KEY,
Duration CHAR(1)
     CONSTRAINT Lease_Duration_NN NOT NULL,
Rental_Address VARCHAR2(300)
     CONSTRAINT Lease_Rental_Address_NN NOT NULL,
Move_Out_Date DATE
     CONSTRAINT Lease_Move_Out_Date_NN NOT NULL,
Move_In_Date DATE
     CONSTRAINT Lease_Move_In_Date_NN NOT NULL,
Room_Number VARCHAR2(7)
     CONSTRAINT Lease_Room_Number_NN NOT NULL,
Phone_Number VARCHAR2(10)
     CONSTRAINT LEASE_Phone_Number_NN NOT NULL
)
```

Lease - Continued

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Table Structure

```
SQL> desc lease
 Name                      Null?    Type
 ------------------------- -------- ----------------
 LEASE_NUMBER              NOT NULL VARCHAR2(5)
 DURATION                  NOT NULL CHAR(1)
 RENTAL_ADDRESS            NOT NULL VARCHAR2(300)
 MOVE_OUT_DATE             NOT NULL DATE
 MOVE_IN_DATE              NOT NULL DATE
 ROOM_NUMBER               NOT NULL VARCHAR2(7)
 PHONE_NUMBER              NOT NULL VARCHAR2(10)
```

Table Contents

```
SQL> select *
  2  from lease;

LEASE_NUMBER DURATION RENTAL_ADDRESS                          MOVE_OUT_DATE MOVE_IN_DATE  ROOM_NUMBER    PHONE_NUMBER
------------ -------- ---------------------------------------- ------------- ------------- -------------- ------------
10000        1        4135 Sierra St. San Bernardino CA. 92408 15-DEC-12     30-AUG-12     3001           9093033030
10001        3        1621 Euerka St. San Bernardino CA. 92407 15-JUN-12     30-AUG-12     3002           9092222222
10002        4        1317 5th St. San Bernardino CA. 92408    15-AUG-12     30-AUG-12     3003           9091111111
10003        2        14 University Ave. San Bernardino CA. 92407 15-MAR-12  30-AUG-12     3004           9093333333
10004        1        74 Euerka St. San Bernardino Ca. 92408   15-DEC-12     30-AUG-12     3005           9094444444
10005        1        123 4th St. San Bernardino CA. 92407     15-DEC-12     30-AUG-12     3006           9097777777
10006        4        14 Market Ave. San Bernardino CA. 92408  15-AUG-12     30-AUG-12     3007           9095555555
10007        4        35 Kinsey Ave. San Bernardino CA. 92407  15-AUG-12     30-AUG-12     3008           9099999999
10008        3        280 Drake Dr. San Bernardino CA. 92408   15-JUN-12     30-AUG-12     3009           9096666666
10009        1        654 Lincoln Ave. San Bernardino Ca. 92408 15-DEC-12    30-AUG-12     3010           9098888888
10010        2        719 Waterman Ave. San Bernardino CA. 92408 15-MAR-12   30-AUG-12     3011           9091234567
10011        1        4564 Waver Ave. San Bernardino Ca. 92408 15-DEC-12     30-AUG-12     3012           9097654321
```

Data Loading: Sql Loader

Description: We used Sql Loader for this table. The ctl and dat files used are below

```
load data
append
into table Lease
fields terminated by "/"
(Lease_Number, Duration, Rental_Address, Move_Out_Date, Move_In_Date, Room_Number, Phone_Number)
```

Lease - Continued

```
10000/1/4135 Sierra St. San Bernardino, CA. 92408/15-DEC-12/30-AUG-12/3001/9093033030
10001/3/1621 Euerka St. San Bernardino, CA. 92407/15-JUN-12/30-AUG-12/3002/9092222222
10002/4/1317 5th St. San Bernardino, CA. 92408/15-AUG-12/30-AUG-12/3003/9091111111
10003/2/14 University Ave. San Bernardino, CA. 92407/15-MAR-12/30-AUG-12/3004/9093333333
10004/1/74 Euerka St. San Bernardino, Ca. 92408/15-DEC-12/30-AUG-12/3005/9094444444
10005/1/123 4th St. San Bernardino, CA. 92407/15-DEC-12/30-AUG-12/3006/9097777777
10006/4/14 Market Ave. San Bernardino, CA. 92408/15-AUG-12/30-AUG-12/3007/9095555555
10007/4/35 Kinsey Ave. San Bernardino, CA. 92407/15-AUG-12/30-AUG-12/3008/9099999999
10008/3/280 Drake Dr. San Bernardino, CA. 92408/15-JUN-12/30-AUG-12/3009/9096666666
10009/1/654 Lincoln Ave. San Bernardino, Ca. 92408/15-DEC-12/30-AUG-12/3010/9098888888
10010/2/719 Waterman Ave. San Bernardino, CA. 92408/15-MAR-12/30-AUG-12/3011/9091234567
10011/1/4564 Waver Ave. San Bernardino, Ca. 92408/15-DEC-12/30-AUG-12/3012/9097654321
```

Staff Member

Team Member Responsible: James Small

Table Comments

TABLE_NAME	TABLE_TYPE	COMMENTS
STAFF_MEMBER	TABLE	A staff member is an employee of the university th at is in charge of a students welfare and academic progress. They are also in charge of inspecting r ooms

Column Comments

TABLE_NAME	COLUMN_NAME	COMMENTS
STAFF_MEMBER	STAFF_NUMBER	The number assigned to the staff at time of hire. This will uniquely identify each staff member - Fo rmat: 1001
	FIRST_NAME	The staff members first name - Format: John
	LAST_NAME	The staff members last name - Format: Smith
	DATE_OF_BIRTH	The staff members date of birth - Format: 13-OCT-8 0
	GENDER	The staff members gender - Format: M
	STREET	The street the staff member lives on - Format: 123 Four St
	CITY	The city the staff member lives in - Format: River side
	STATE	The state the staff member lives in - Format: CA
	ZIP	The zip code the staff member lives in - Format: 9 2504
	LOCATION	The location where the staff member works in the u niversity - Format: CSUSB Main
	POSITION	The staff members position at the university - For mat: CSE Professor
	EMAIL_ADDRESS	The staff members email address- Format: agonz@csu sb.edu

Staff Member - Continued

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: James Small
*/

CREATE TABLE Staff_Member (
Staff_Number CHAR(4)
      CONSTRAINT Staff_Member_PK PRIMARY KEY,
First_Name VARCHAR2(30)
      CONSTRAINT Staff_Member_First_Name_NN NOT NULL,
Last_Name VARCHAR2(30)
      CONSTRAINT Staff_Member_Last_Name_NN NOT NULL,
Date_of_Birth DATE DEFAULT SYSDATE
      CONSTRAINT Staff_Member_Date_of_Birth_NN NOT NULL,
Gender CHAR(1)
      CONSTRAINT Staff_Member_Gender_NN NOT NULL
      CONSTRAINT Staff_Member_Gender_CK CHECK (Gender in ('F','f','M','m')),
Street VARCHAR2(30)
      CONSTRAINT Staff_Member_Street_NN NOT NULL,
City VARCHAR2(30)
      CONSTRAINT Staff_Member_City_NN NOT NULL,
State CHAR(2)
      CONSTRAINT Staff_Member_State_NN NOT NULL,
Zip VARCHAR2(10)
      CONSTRAINT Staff_Member_Zip_NN NOT NULL,
Location VARCHAR2(30)
      CONSTRAINT Staff_Member_Location_NN NOT NULL,
Position VARCHAR2(30)
      CONSTRAINT Staff_Member_Position_NN NOT NULL,
Email_Address VARCHAR2(60)
      CONSTRAINT Staff_Member_Email_CK CHECK (email_address like '%@%')
)
```

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Staff Member - Continued

Table Structure

```
SQL> desc staff_member
 Name                      Null?    Type
 ------------------------- -------- ----------------
 STAFF_NUMBER              NOT NULL CHAR(4)
 FIRST_NAME                NOT NULL VARCHAR2(30)
 LAST_NAME                 NOT NULL VARCHAR2(30)
 DATE_OF_BIRTH             NOT NULL DATE
 GENDER                    NOT NULL CHAR(1)
 STREET                    NOT NULL VARCHAR2(30)
 CITY                      NOT NULL VARCHAR2(30)
 STATE                     NOT NULL CHAR(2)
 ZIP                       NOT NULL VARCHAR2(10)
 LOCATION                  NOT NULL VARCHAR2(30)
 POSITION                  NOT NULL VARCHAR2(30)
 EMAIL_ADDRESS                      VARCHAR2(60)
```

Table Contents

```
SQL> select *
  2  from staff_member;

STAFF_NUMBER FIRST_NAME   LAST_NAME    DATE_OF_BIRTH GENDER STREET          CITY            STATE ZIP   LOCATION    POSITION          EMAIL_ADDRESS
------------ ------------ ------------ ------------- ------ --------------- --------------- ----- ----- ----------- ----------------- --------------------
1000         Anthony      Gonzalez     06-MAY-47     M      123 Four St     Redlands        CA    92373 CSUSB Main  Phys Ed Professor agonz@csusb.edu
1001         Michelle     Smith        09-JUN-58     F      654 Fourth St   Riverside       CA    92504 CSUSB Main  CSE Professor     msmith@csusb.edu
1002         Mark         Preciado     03-JAN-45     M      352 Anderson Way Riverside      CA    92504 CSUSB Main  Math Tutor        mpreciado@csusb.edu
1003         Ashley       Lopez        03-APR-73     F      984 Halsey St   Redlands        CA    92375 CSUSB Main  English Professor alopez@csusb.edu
1004         Sergio       Martinez     11-DEC-82     M      253 Wilson Way  San Bernardino  CA    92408 CSUSB Main  Math Professor    smartinez@csusb.edu
1005         Lisa         Turtle       27-SEP-61     F      352 Bell Way    Chino           CA    92347 CSUSB Main  History Professor lturtle@csusb.edu
```

Data Loading: Normal Insert

Description: We used the normal insert method for this table. The insert script is below.

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
Data Load Method 1: Normal Insert - Insert SQL Command
Done By: James Small
*/

INSERT INTO  staff_member(Staff_number, First_name,Last_Name,Date_Of_Birth,Gender,Street,City,State,Zip,Location,Position,Email_address)
values ('1000','Anthony', 'Gonzalez','06-MAY-47','M','123 Four St','Redlands','CA','92373','CSUSB Main','Phys Ed Professor','agonz@csusb.edu');

INSERT INTO  staff_member(Staff_number, First_name,Last_Name,Date_Of_Birth,Gender,Street,City,State,Zip,Location,Position,Email_address)
values ('1001','Michelle', 'Smith','09-JUN-58','F','654 Fourth St','Riverside','CA','92504','CSUSB Main','CSE Professor','msmith@csusb.edu');

INSERT INTO  staff_member(Staff_number, First_name,Last_Name,Date_Of_Birth,Gender,Street,City,State,Zip,Location,Position,Email_address)
values ('1002','Mark', 'Preciado','03-JAN-45','M','352 Anderson Way','Riverside','CA','92504','CSUSB Main','Math Tutor','mpreciado@csusb.edu');

INSERT INTO  staff_member(Staff_number, First_name,Last_Name,Date_Of_Birth,Gender,Street,City,State,Zip,Location,Position,Email_address)
values ('1003','Ashley', 'Lopez','03-APR-73','F','984 Halsey St','Redlands','CA','92375','CSUSB Main','English Professor','alopez@csusb.edu');

INSERT INTO  staff_member(Staff_number, First_name,Last_Name,Date_Of_Birth,Gender,Street,City,State,Zip,Location,Position,Email_address)
values ('1004','Sergio', 'Martinez','11-DEC-82','M','253 Wilson Way','San Bernardino','CA','92408','CSUSB Main','Math Professor','smartinez@csusb.edu');

INSERT INTO  staff_member(Staff_number, First_name,Last_Name,Date_Of_Birth,Gender,Street,City,State,Zip,Location,Position,Email_address)
values ('1005','Lisa', 'Turtle','27-SEP-61','F','352 Bell Way','Chino','CA','92347','CSUSB Main','History Professor','lturtle@csusb.edu');
```

Advisor

Team Member Responsible: Daniel Urbach

Table Comments

```
TABLE_NAME           TABLE_TYPE COMMENTS
-------------------- ---------- --------------------------------------------------
ADVISOR              TABLE      The Advisor is a staff-member who advises students
                                 in the database on academic progress
```

Column Comments

```
TABLE_NAME           COLUMN_NAME                  COMMENTS
-------------------- ---------------------------- --------------------------------------------------
ADVISOR              ADVISOR_ROOM_NUMBER          Office room number assigned to Advisor - format: P
                                                  S401

                     PHONE_NUMBER                 Advisors primary phone number - format: 5551234567
                     DEPARTMENT                   Department to which the Advisor belongs to - forma
                                                  t: History

                     ADVISOR_NUMBER               Foreign key corresponding to the Advisor-specializ
                                                  ation of Staff Member - format: 1001
```

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Daniel Urbach
*/

CREATE TABLE Advisor (
Advisor_Room_Number VARCHAR2(10)
      CONSTRAINT Advisor_Room_Number_NN NOT NULL,
Phone_Number VARCHAR2(10)
      CONSTRAINT Advisor_Phone_Number_NN NOT NULL,
Department VARCHAR2(30)
      CONSTRAINT Advisor_Department_NN NOT NULL,
Advisor_Number CHAR(4)
      CONSTRAINT Advisor_PK PRIMARY KEY,
CONSTRAINT Advisor_Advisor_Number_FK
      FOREIGN KEY (Advisor_Number)
      REFERENCES Staff_Member (Staff_Number) ON DELETE CASCADE
)
```

Advisor - Continued

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Table Structure

```
SQL> desc advisor
 Name                      Null?    Type
 ------------------------- -------- -----------------
 ADVISOR_ROOM_NUMBER       NOT NULL VARCHAR2(10)
 PHONE_NUMBER              NOT NULL VARCHAR2(10)
 DEPARTMENT                NOT NULL VARCHAR2(30)
 ADVISOR_NUMBER            NOT NULL CHAR(4)
```

Table Contents

```
SQL> select *
  2  from advisor;

ADVISOR_ROOM_NUMBER  PHONE_NUMBER   DEPARTMENT           ADVISOR_NUMBER
-------------------- -------------- -------------------- --------------------
PH102                9095551234     Physical Education   1001
UH401                9095556789     History              1003
UH222                9095377598     CSE                  1005
SB123                9094567845     Political Science    1000
```

Data Loading: Normal Insert

Description: We used the normal insert method for this table. The insert script is below.

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
Data Load Method 1: Normal Insert - Insert SQL Command
Done By: Daniel Urbach
*/

INSERT INTO  advisor(advisor_room_number, phone_number, department, advisor_number)
values ('PH102','9095551234','Physical Education','1001');
INSERT INTO  advisor(advisor_room_number, phone_number, department, advisor_number)
values ('UH401','9095556789','History','1003');
INSERT INTO  advisor(advisor_room_number, phone_number, department, advisor_number)
values ('UH222','9095377598','CSE','1005');
INSERT INTO  advisor(advisor_room_number, phone_number, department, advisor_number)
values ('SB123','9094567845','Political Science','1000');
```

Next of Kin

Team Member Responsible: Enrique Carbajal

Table Comments

TABLE_NAME	TABLE_TYPE	COMMENTS
NEXT_OF_KIN	TABLE	This attribute is the person to contact in case of emergency. This is a composite attribute. Its composition is comprised of relationship to student, the persons name, phone number and address. Address is also a composite attribute.

Column Comments

TABLE_NAME	COLUMN_NAME	COMMENTS
NEXT_OF_KIN	NOK_NAME	This is the next of kins name--Format: Enrique Carbajal Jr.
	NOK_PHONE_NUMBER	This is the NOKs phone number --Format: 995-846-7778
	NOK_STREET	This is the NOKs street address --Format: 987 Alpha St
	NOK_CITY	This is the NOKs current resident city --Format: Riverside
	NOK_STATE	This is the NOKs current resident State --Format: CA
	NOK_ZIP	This is the NOKs current resident zip code --Format: 92504

Next_of_Kin - Continued

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Enrique Carbajal
*/

CREATE TABLE Next_of_Kin (
NOK_Name VARCHAR2(100)
     CONSTRAINT NOK_PK PRIMARY KEY,
NOK_Phone_Number VARCHAR2(10)
     CONSTRAINT NOK_NOK_Phone_Number_NN NOT NULL,
NOK_Street VARCHAR2(30)
     CONSTRAINT NOK_NOK_Street_NN NOT NULL,
NOK_City VARCHAR2(30)
     CONSTRAINT NOK_City_NN NOT NULL,
NOK_State CHAR(2)
     CONSTRAINT NOK_State_NN NOT NULL,
NOK_Zip VARCHAR2(10)
     CONSTRAINT NOK_NOK_Zip_NN NOT NULL
)
```

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Table Structure

```
SQL> desc next_of_kin
 Name                      Null?    Type
 ------------------------- -------- ----------------
 NOK_NAME                  NOT NULL VARCHAR2(100)
 NOK_PHONE_NUMBER          NOT NULL VARCHAR2(10)
 NOK_STREET                NOT NULL VARCHAR2(30)
 NOK_CITY                  NOT NULL VARCHAR2(30)
 NOK_STATE                 NOT NULL CHAR(2)
 NOK_ZIP                   NOT NULL VARCHAR2(10)
```

Next_of_Kin - Continued

Table Contents

```
SQL> select *
  2  from next_of_kin;

NOK_NAME            NOK_PHONE_NUMBER NOK_STREET          NOK_CITY          NO NOK_ZIP
------------------- ---------------- ------------------- ----------------- -- ----------
Mark Smith          9958467778       987 Alpha St        Riverside         CA 92504
Tony Stark          9094567815       1534 Anderson Ave   Redlands          CA 92373
Robert Mitts        6546547895       3564 Besson Lane    Chino             CA 92445
William Tell        9515647914       333 Citrus Ave      Ontario           CA 92446
Art Inder           6654457921       66 Holt Ave         San Bernardino    CA 92408
Carl Vaughn         2144676657       35335 Willow Way    Upland            CA 92444
Scott Nelson        2146559998       3374 Sierra Ave     Pomona            CA 92437
Ashley Doherty      3153214556       674 Arch Way        Big Bear          CA 92404
Terry Spinks        3366547779       353 Gardena Ave     Adelanto          CA 92616
```

Data Loading: Substitution

Description: We used the substitution method for this table. The substitution script is below.

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
Data Load Method 2: Insert Using Substitution Variables
Done By: Enrique Carbajal Jr.
*/

INSERT INTO Next_of_Kin ( NOK_Name, NOK_Phone_Number, NOK_Street, NOK_City, NOK_State, NOK_Zip)
values( '&NOK_Name', '&NOK_Phone_Number', '&NOK_Street', '&NOK_City', '&NOK_State', '&NOK_Zip');
```

Student

Team Member Responsible: James Small

Table Comments

TABLE_NAME	TABLE_TYPE	COMMENTS
STUDENT	TABLE	A student is an individual who is currently enrolled in the university that is living in a residence or on the waiting list

Column Comments

TABLE_NAME	COLUMN_NAME	COMMENTS
STUDENT	STUDENT_NUMBER	The number that uniquely identifies each student at the university - Format: 600001
	FIRST_NAME	The first name of the student - Format: John
	LAST_NAME	The last name of the student - Format: Smith
	GENDER	The gender of the student - Format: M
	DATE_OF_BIRTH	The birthdate for the student - Format: 05-JAN-85
	CURRENT_STATUS	Shows whether a student is currently renting or on the waitlist - Format: Renting
	ADDITIONAL_COMMENTS	Any additional information pertaining to the student not mentioned elsewhere - Format: Text
	RESIDENCE_TYPE	Shows the type of residence the student is currently renting - Format: Student Apartment
	MINOR	The students minor if exists - Format: Philosphy
	MAJOR	The students major at the university- Format: Philosphy
	SPECIAL_NEEDS	Information in this field lets us know whether there is any special accomodations needed for the student - Format: Blind
	STREET	The street the student lives on - Format: 123 Four St
	CITY	The city the student lives in - Format: Riverside
	STATE	The state the student lives in - Format: CA
	ZIP	The zip code the student lives in - Format: 92504
	NATIONALITY	The nation of origin or naturalization of student - Format: German
	CLASS	The standing of student in terms of academic progress - Format: Junior
	PHONE_NUMBER	The phone number of the student, if applicable - Format: 909-844-8887
	EMAIL_ADDRESS	The email address of the student- Format: agonz@gmail.com
	ADVISOR_NUMBER	Foreign key referencing advisor_number in advisor. Shows which advisor is responsible for advising this particular student - Format: 1001

Student - Continued

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: James Small
*/

CREATE TABLE Student (
Student_Number VARCHAR2(6)
      CONSTRAINT Student_PK PRIMARY KEY,
First_Name VARCHAR2(30)
      CONSTRAINT Student_First_Name_NN NOT NULL,
Last_Name VARCHAR2(30)
      CONSTRAINT Student_Last_Name_NN NOT NULL,
Gender CHAR(1)
      CONSTRAINT Student_Gender_NN NOT NULL,
Date_of_Birth DATE
      CONSTRAINT Student_Date_of_Birth_NN NOT NULL,
Current_Status VARCHAR2(7)
      CONSTRAINT Student_Current_Status_NN NOT NULL,
Additional_Comments VARCHAR2(4000),
Residence_Type VARCHAR2(17),
Minor VARCHAR2(30),
Major VARCHAR2(30)
      CONSTRAINT Student_Major_NN NOT NULL,
Special_Needs VARCHAR2(4000),
      Street VARCHAR2(30)
      CONSTRAINT Student_Street_NN NOT NULL,
City VARCHAR2(30)
      CONSTRAINT Student_City_NN NOT NULL,
State CHAR(2)
      CONSTRAINT Student_State_NN NOT NULL,
Zip VARCHAR2(10)
      CONSTRAINT Student_Zip_NN NOT NULL,
Nationality VARCHAR2(30),
Class VARCHAR2(12)
      CONSTRAINT Student_Class_NN NOT NULL,
Phone_Number VARCHAR2(10),
Email_Address VARCHAR2(60),
Advisor_Number CHAR(4)
      CONSTRAINT Student_Advisor_Number_NN NOT NULL,
CONSTRAINT Student_Advisor_Number_FK
      FOREIGN KEY (Advisor_Number)
      REFERENCES Advisor (Advisor_Number) ON DELETE SET NULL
)
```

Student - Continued

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table

commands to add constraints. The data insertion also followed this same order.

Table Structure

```
SQL> desc student
 Name                      Null?    Type
 ------------------------- -------- ----------------
 STUDENT_NUMBER            NOT NULL VARCHAR2(6)
 FIRST_NAME                NOT NULL VARCHAR2(30)
 LAST_NAME                 NOT NULL VARCHAR2(30)
 GENDER                    NOT NULL CHAR(1)
 DATE_OF_BIRTH             NOT NULL DATE
 CURRENT_STATUS            NOT NULL VARCHAR2(7)
 ADDITIONAL_COMMENTS                VARCHAR2(4000)
 RESIDENCE_TYPE                     VARCHAR2(17)
 MINOR                              VARCHAR2(30)
 MAJOR                     NOT NULL VARCHAR2(30)
 SPECIAL_NEEDS                      VARCHAR2(4000)
 STREET                    NOT NULL VARCHAR2(30)
 CITY                      NOT NULL VARCHAR2(30)
 STATE                     NOT NULL CHAR(2)
 ZIP                       NOT NULL VARCHAR2(10)
 NATIONALITY                        VARCHAR2(30)
 CLASS                     NOT NULL VARCHAR2(12)
 PHONE_NUMBER                       VARCHAR2(10)
 EMAIL_ADDRESS                      VARCHAR2(60)
 ADVISOR_NUMBER            NOT NULL CHAR(4)
```

Table Contents

```
SQL> select *
  2  from student;

STUDENT_NUMBER FIRST_NAME   LAST_NAME    GENDER DATE_OF_BIRTH CURRENT ADDITIONAL_COMMENTS RESIDENCE_TYPE    MINOR            MAJOR            SPECIAL_NEEDS
-------------- ------------ ------------ ------ ------------- ------- -------------------- ---------------- ---------------- ---------------- --------------
600001         Curtis       Wright       M      10-JUL-80     Renting None                 Residence Hall   Math             Computer Science None
600002         Michael      Baldridge    M      04-SEP-82     Renting None                 Residence Hall                    Math             None
600003         Liz          Smith        F      11-JUN-87     Renting None                 Residence Hall                    English          Wheelchair
600004         Andrew       Sherwood     M      11-DEC-91     Renting None                 Residence Hall   Finance          Computer Systems None
600007         Jennifer     West         F      12-APR-90     Renting None                 Residence Hall                    Education        None
600012         Paul         Darkwa       M      01-JAN-81     Waiting None                                                   Psychology       Wheelchair
600014         Melanie      Williams     F      03-MAR-83     Renting None                 Residence Hall                    Physics          None
600005         Melissa      Vaughn       F      01-DEC-93     Renting None                 Student Apartment Math            Computer Science None
600006         Beth         Lopez        F      04-MAR-89     Renting None                 Student Apartment Physics         Computer Science None
600008         John         Wiley        M      11-JUL-77     Renting None                 Student Apartment Math            Physics          None
600009         Jessica      Wilson       F      01-AUG-88     Renting None                 Student Apartment                 Art History      None
600010         Ryan         Oliver       M      27-SEP-89     Renting None                 Student Apartment                 Liberal Studies  Deaf
600011         Jesse        Gomez        M      11-APR-92     Renting None                 Student Apartment                 Philosphy        None
600013         Susan        Storey       F      02-FEB-82     Waiting None                                   Computer Science Communications   Blind
```

Student - Continued

STREET	CITY	STATE	ZIP	NATIONALITY	CLASS	PHONE_NUMBER	EMAIL_ADDRESS	ADVISOR_NUMBER
456 Besson Lane	Redlands	CA	92373	American	Freshman	9091111111	cwright@coyote.csusb.edu	1001
457 Wilshire Blvd	Grand Terrace	CA	92409	Austrailian	Sophomore	9092222222	mbaldridge@coyote.csusb.edu	1003
41 Rock Blvd	Chino	CA	92647	American	Junior	9093333333	lsmith@coyote.csusb.edu	1005
3654 Brookside	Redlands	CA	92374	Japanese	Freshman	9094444444	asherwood@coyote.csusb.edu	1000
35 Pocket St	Beaumont	CA	92222	American	Junior	9097777777	jwest@coyote.csusb.edu	1000
3634 Torrent Rd	Ontario	CA	92665	American	Freshman	9091011010	pdarkwa@coyote.csusb.edu	1005
896 Asus St	Loma Linda	CA	92377	American	Freshman	9093033030	mwilliams@coyote.csusb.edu	1001
253 Sierra St	Riverside	CA	92504	American	Senior	9095555555	mvaughn@coyote.csusb.edu	1003
2533 Wilson Ave	Banning	CA	92223	American	Sophomore	9096666666	blopez@coyote.csusb.edu	1001
30566 Ash St	San Bernardino	CA	92408	American	Freshman	9098888888	jwiley@coyote.csusb.edu	1005
1400 Barton Rd	Arrowhead	CA	92404	Korean	Senior	9099999999	jwilson@coyote.csusb.edu	1000
30 N Fish St	Pomona	CA	92746	German	Sophomore	9091234567	roliver@coyote.csusb.edu	1003
135 Wilshire Blvd	Chino	CA	92465	Italian	Junior	9097654321	jgomez@coyote.csusb.edu	1001
3260 Peak St	Rialto	CA	92344	American	Senior	9092022020	sstorey@coyote.csusb.edu	1003

Data Loading: Sql Loader

Description: We used Sql Loader for this table. The ctl and dat files used are below

```
load data
append
into table student
fields terminated by "/"
(student_number,first_name,last_name,gender,date_of_birth,current_status,additional_comments,residence_type,minor,major,special_needs,
     street,city,state,zip,nationality,class,phone_number,email_address,advisor_number)
```

Note: Last line of ctl file was shown on two lines for readability. Real file was all on one line.

```
600001/Curtis/Wright/M/10-JUL-80/Renting/None/Residence Hall/Math/Computer Science/None/456 Besson Lane/Redlands/CA/92373/American/Freshman/9091111111/cwright@coyote.csusb.edu/1001
600002/Michael/Baldridge/M/04-SEP-82/Renting/None/Residence Hall//Math/None/457 Wilshire Blvd/Grand Terrace/CA/92409/Austrailian/Sophomore/9092222222/mbaldridge@coyote.csusb.edu/1003
600003/Liz/Smith/F/11-JUN-87/Renting/None/Residence Hall//English/Wheelchair/41 Rock Blvd/Chino/CA/92647/American/Junior/9093333333/lsmith@coyote.csusb.edu/1005
600004/Andrew /Sherwood/M/11-DEC-91/Renting/None/Residence Hall/Finance/Computer Systems/None/3654 Brookside/Redlands/CA/92374/Japanese/Freshman/9094444444/asherwood@coyote.csusb.edu/1000
600005/Melissa/Vaughn/F/01-DEC-93/Renting/None/Student Apartment/Math/Computer Science/None/253 Sierra St/Riverside/CA/92504/American/Senior/9095555555/mvaughn@coyote.csusb.edu/1003
600006/Beth/Lopez/F/04-MAR-89/Renting/None/Student Apartment/Physics/Computer Science/None/2533 Wilson Ave/Banning/CA/92223/American/Sophomore/9096666666/blopez@coyote.csusb.edu/1001
600007/Jennifer/West/F/12-APR-90/Renting/None/Residence Hall//Education/None/35 Pocket St/Beaumont/CA/92222/American/Junior/9097777777/jwest@coyote.csusb.edu/1000
600008/John/Wiley/M/11-JUL-77/Renting/None/Student Apartment/Math/Physics/None/30566 Ash St/San Bernardino/CA/92408/American/Freshman/9098888888/jwiley@coyote.csusb.edu/1005
600009/Jessica/Wilson/F/01-AUG-88/Renting/None/Student Apartment//Art History/None/1400 Barton Rd/Arrowhead/CA/92404/Korean/Senior/9099999999/jwilson@coyote.csusb.edu/1000
600010/Ryan/Oliver/M/27-SEP-89/Renting/None/Student Apartment//Liberal Studies/Deaf/30 N Fish St/Pomona/CA/92746/German/Sophomore/9091234567/roliver@coyote.csusb.edu/1003
600011/Jesse/Gomez/M/11-APR-92/Renting/None/Student Apartment//Philosphy/None/135 Wilshire Blvd/Chino/CA/92465/Italian/Junior/9097654321/jgomez@coyote.csusb.edu/1001
600012/Paul/Darkwa/M/01-JAN-81/Waiting/None///Psychology/Wheelchair/3634 Torrent Rd/Ontario/CA/92665/American/Freshman/9091011010/pdarkwa@coyote.csusb.edu/1005
600013/Susan/Storey/F/02-FEB-82/Waiting/None//Computer Science/Communications/Blind/3260 Peak St/Rialto/CA/92344/American/Senior/9092022020/sstorey@coyote.csusb.edu/1003
600014/Melanie/Williams/F/03-MAR-83/Renting/None/Residence Hall//Physics/None/896 Asus St/Loma Linda/CA/92377/American/Freshman/9093033030/mwilliams@coyote.csusb.edu/1001
```

NOK_Relationship

Team Member Responsible: James Small

Table Comments

TABLE_NAME	TABLE_TYPE	COMMENTS
NOK_RELATIONSHIP	TABLE	The relationship between a student in the database and their Next of Kin

Column Comments

TABLE_NAME	COLUMN_NAME	COMMENTS
NOK_RELATIONSHIP	STUDENT_NUMBER	Foreign key referencing Student Number in Student relation, part of the primary key - format: 123456
	NOK_NAME	Foreign key referencing NOK Name in Next of Kin relation, part of the primary key - format: Jimmy Dean
	NOK_RELATIONSHIP	The relationship between a student and the next of kin - format: Brother

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: James Small
*/

CREATE TABLE NOK_Relationship (
Student_Number VARCHAR2(6)
     CONSTRAINT nok_relationship_studentno_FK REFERENCES Student(Student_number) ON DELETE CASCADE,
NOK_Name VARCHAR2(100)
     CONSTRAINT nok_relationship_name_FK REFERENCES Next_of_Kin(NOK_Name) ON DELETE CASCADE,
NOK_Relationship VARCHAR2(30)
     CONSTRAINT NOK_NOK_Relationship_NN NOT NULL,
CONSTRAINT nok_relationship_PK PRIMARY KEY (student_number,nok_name)
)
```

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

NOK_Relationship - Continued

Table Structure

```
SQL> desc NOK_relationship
 Name                      Null?    Type
 ------------------------- -------- ----------------
 STUDENT_NUMBER            NOT NULL VARCHAR2(6)
 NOK_NAME                  NOT NULL VARCHAR2(100)
 NOK_RELATIONSHIP          NOT NULL VARCHAR2(30)
```

Table Contents

```
SQL> select *
  2  from nok_relationship;

STUDENT_NUMBER  NOK_NAME              NOK_RELATIONSHIP
--------------- -------------------- ----------------------
600001          Mark Smith           Brother
600002          Tony Stark           Father
600010          Robert Mitts         Brother
600014          William Tell         Uncle
600012          Art Inder            Brother
600004          Carl Vaughn          Father
600008          Scott Nelson         Cousin
600013          Ashley Doherty       Sister
600006          Terry Spinks         Mother
```

Data Loading: Sql Loader

Description: We used Sql Loader for this table. The ctl and dat files used are below

```
load data
append
into table NOK_Relationship
fields terminated by "/"
(student_number,NOK_Name,NOK_Relationship)
```

NOK_Relationship - Continued

```
600001/Mark Smith/Brother
600002/Tony Stark/Father
600010/Robert Mitts/Brother
600014/William Tell/Uncle
600012/Art Inder/Brother
600004/Carl Vaughn/Father
600008/Scott Nelson/Cousin
600013/Ashley Doherty/Sister
600006/Terry Spinks/Mother
```

Instructor

Team Member Responsible: Daniel Urbach

Table Comments

```
TABLE_NAME           TABLE_TYPE COMMENTS
-------------------- ---------- --------------------------------------------------
INSTRUCTOR           TABLE      Instructor is the person teaching a course being t
                                aken by students in the database
```

Column Comments

```
TABLE_NAME           COLUMN_NAME                    COMMENTS
-------------------- ------------------------------ --------------------------------------------------
INSTRUCTOR           INSTRUCTOR_NAME                Name of Instructor teaching a Course - format: Jos
                                                    ephine Mendoza

                     INSTRUCTOR_EMAIL               Email address for Instructor - format: jmendoza@cs
                                                    usb.edu

                     INSTRUCTOR_PHONE               Phone number for Instructor - format: 90952773822
```

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Danny Urbach
*/

CREATE TABLE Instructor (
Instructor_Name VARCHAR2(100)
     CONSTRAINT Instructor_PK PRIMARY KEY,
Instructor_Email VARCHAR2(60)
     CONSTRAINT Instructor_Email_NN NOT NULL,
Instructor_Phone VARCHAR2(10)
     CONSTRAINT Instructor_Phone_NN NOT NULL
)
```

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Instructor - Continued

Table Structure

```
SQL> desc instructor
 Name                      Null?    Type
 ------------------------- -------- ----------------
 INSTRUCTOR_NAME           NOT NULL VARCHAR2(100)
 INSTRUCTOR_EMAIL          NOT NULL VARCHAR2(60)
 INSTRUCTOR_PHONE          NOT NULL VARCHAR2(10)
```

Table Contents

```
SQL> select *
  2  from instructor;

INSTRUCTOR_NAME           INSTRUCTOR_EMAIL               INSTRUCTOR_PHONE
------------------------- ------------------------------ --------------------
Richard Botting           rbotting@csusb.edu             4567891230
Josephine Mendoza         jmendoza@csusb.edu             6543219870
Owen Murphy               omurphy@csusb.edu              9876543210
Kay Zemoudeh              kzemoudeh@csusb.edu            3031233030
Arturo Concepcion         aconcepcion@csusb.edu          6654567787
Yasha Karant              ykarant@csusb.edu              2146664524
```

Data Loading: Sql Loader

Description: We used Sql Loader for this table. The ctl and dat files used are below

```
load data
append
into table instructor
fields terminated by "/"
(instructor_name,instructor_email,instructor_phone)
```

```
Richard Botting/rbotting@csusb.edu/4567891230
Josephine Mendoza/jmendoza@csusb.edu/6543219870
Owen Murphy/omurphy@csusb.edu/9876543210
Kay Zemoudeh/kzemoudeh@csusb.edu/3031233030
Arturo Concepcion/aconcepcion@csusb.edu/6654567787
Yasha Karant/ykarant@csusb.edu/2146664524
```

Course

Team Member Responsible: Daniel Urbach

Table Comments

```
TABLE_NAME           TABLE_TYPE COMMENTS
-------------------- ---------- --------------------------------------------------
COURSE               TABLE      Course is a class taken by students in the databas
                                e, offered by the university
```

Column Comments

```
TABLE_NAME           COLUMN_NAME                    COMMENTS
-------------------- ------------------------------ --------------------------------------------------
COURSE               COURSE_NUMBER                  The unique identifer for Course acting as primary
                                                    key - format: CSE572

                     TITLE                          The descriptive name of the Course - format: Datab
                                                    ase Systems

                     ROOM_NUMBER                    The location where the Course is conducted - forma
                                                    t: JB359

                     DEPARTMENT                     The department which the Course belongs to - forma
                                                    t: Computer Science

                     INSTRUCTOR_NAME                Foreign key referencing Instructor_Name in the Ins
                                                    tructor relation, name of the instructor teaching
                                                    the Course - format: Josephine Mendoza
```

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Daniel Urbach
*/

CREATE TABLE Course (
Course_Number VARCHAR2(7)
      CONSTRAINT Course_PK PRIMARY KEY,
Title VARCHAR2(30)
      CONSTRAINT Course_Title_NN NOT NULL,
Room_Number VARCHAR2(10)
      CONSTRAINT Course_Room_Number_NN NOT NULL,
Department VARCHAR2(30)
      CONSTRAINT Course_Department_NN NOT NULL,
Instructor_Name VARCHAR2(100)
      CONSTRAINT Course_Instructor_Name_NN NOT NULL,
CONSTRAINT Course_Instructor_Name_FK
      FOREIGN KEY (Instructor_Name)
      REFERENCES Instructor(Instructor_Name) ON DELETE SET NULL
)
```

Course - Continued

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Table Structure

```
SQL> desc course
 Name                      Null?    Type
 ------------------------- -------- ----------------
 COURSE_NUMBER             NOT NULL VARCHAR2(7)
 TITLE                     NOT NULL VARCHAR2(30)
 ROOM_NUMBER               NOT NULL VARCHAR2(10)
 DEPARTMENT                NOT NULL VARCHAR2(30)
 INSTRUCTOR_NAME           NOT NULL VARCHAR2(100)
```

Table Contents

```
SQL> select *
  2  from course;

COURSE_NUMBER TITLE                        ROOM_NUMBER     DEPARTMENT          INSTRUCTOR_NAME
------------- ---------------------------- --------------- ------------------- ----------------------
CSE572        Database Systems             PS110           Computer Science    Josephine Mendoza
CSE330        Data Structures              JB359           Computer Science    Kay Zemoudeh
CSE201        Computer Science I           JB358           Computer Science    Richard Botting
CSE202        Computer Science II          JB360           Computer Science    Richard Botting
CSE455        Software Engineering         JB120           Computer Science    Arturo Concepcion
CSE488        Ethics                       UH120           Computer Science    Yasha Karant
CSE350        File Systems                 UH220           Computer Science    Owen Murphy
Comm311       Business Communications      UH350           Communication       Yasha Karant
CSE460        Operating Systems            JB114           Computer Science    Kay Zemoudeh
CSE580        Advanced Databases           JB220           Computer Science    Josephine Mendoza
```

Data Loading: Substitution

Description: We used the substitution method for this table. The substitution script is below.

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
Data Load Method 2: Insert Using Substitution Variables
Done By: Daniel Urbach
*/

INSERT INTO  course(Course_number, Title, Room_number, Department, Instructor_name)
values ('&Course_number', '&Title','&Room_number','&Department','&Instructor_name');
```

Takes_Course

Team Member Responsible: James Small

Table Comments

TABLE_NAME	TABLE_TYPE	COMMENTS
TAKES_COURSE	TABLE	Takes Course defines which students are taking which courses at the university

Column Comments

TABLE_NAME	COLUMN_NAME	COMMENTS
TAKES_COURSE	STUDENT_NUMBER	Foreign key referencing student_number in Student relation. Used as part of the composite primary key to represent a course a student is taking - Format: 6001
	COURSE_NUMBER	Foreign key referencing course_number in Course relation. Used as part of the composite primary key to represent a course a student is taking - Format: CSE572

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: James Small
*/

CREATE TABLE Takes_Course (
Student_Number VARCHAR2(6),
Course_Number VARCHAR2(7),
CONSTRAINT Takes_Course_PK
      PRIMARY KEY (Student_Number, Course_Number),
CONSTRAINT Takes_Course_Student_Number_FK
      FOREIGN KEY (Student_Number)
      REFERENCES Student (Student_Number) ON DELETE CASCADE,
CONSTRAINT Takes_Course_Course_Number_FK
      FOREIGN KEY (Course_Number)
      REFERENCES Course (Course_Number) ON DELETE CASCADE
)
```

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Takes_Course - Continued

Table Structure

```
SQL> desc takes_course
 Name                      Null?    Type
 ------------------------- -------- ----------------
 STUDENT_NUMBER            NOT NULL VARCHAR2(6)
 COURSE_NUMBER             NOT NULL VARCHAR2(7)
```

Table Contents

```
SQL> select *
  2  from takes_course;

STUDENT_NUMBER  COURSE_NUMBER
--------------- ---------------
600001          CSE455
600001          CSE572
600002          CSE330
600002          CSE488
600003          CSE202
600003          CSE488
600004          CSE350
600004          CSE460
600005          CSE460
600005          CSE580
600006          CSE488
600006          Comm311
600007          CSE330
600007          CSE572
600008          CSE350
600008          Comm311
600009          CSE350
600009          CSE460
600010          CSE455
600010          CSE580
600011          CSE460
600011          CSE580
600012          CSE330
600012          Comm311
600013          CSE488
600013          CSE580
600014          CSE201
```

Takes_Course - Continued

Data Loading: Substitution

Description: We used the substitution method for this table. The substitution script is below.

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
Data Load Method 2: Insert Using Substitution Variables
Done By: James Small
*/

INSERT INTO  takes_course(student_number,course_number)
values ('&student_number','&course_number');
```

Rented_Residence

Team Member Responsible: Mark Takahashi

Table Comments

```
TABLE_NAME           TABLE_TYPE COMMENTS
-------------------- ---------- --------------------------------------------------
RENTED_RESIDENCE     TABLE      A Rented Residence is the current location of a ro
                                om that is being occupied by a student
```

Column Comments

```
TABLE_NAME           COLUMN_NAME                    COMMENTS
-------------------- ------------------------------ --------------------------------------------------
RENTED_RESIDENCE     RESIDENCE_NUMBER                Primary key identifying the rented residence -- F
                                                    ormat: 2001

                     ADDRESS                        The address of the residence -- Format: 4135 Sierr
                                                    a St. San Bernardino, CA. 92408
```

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Mark Takahashi
*/

CREATE TABLE Rented_Residence (
Residence_Number CHAR(4)
     CONSTRAINT Rented_Residence_PK PRIMARY KEY,
Address VARCHAR2(200)
     CONSTRAINT Rented_Residence_Address_NN NOT NULL
)
```

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Rented_Residence - Continued

Table Structure

```
SQL> desc rented_residence
 Name                      Null?    Type
 ------------------------- -------- ----------------
 RESIDENCE_NUMBER          NOT NULL CHAR(4)
 ADDRESS                   NOT NULL VARCHAR2(200)
```

Table Contents

```
SQL> select *
  2  from rented_residence;

RESIDENCE_NUMBER     ADDRESS
-------------------- --------------------------------------------------
2001                 4135 Sierra St. San Bernardino, CA. 92408
2002                 1621 Euerka St. San Bernardino, CA. 92407
2003                 1317 5th St. San Bernardino, CA. 92408
2004                 14 University Ave, San Bernardino, CA. 92407
2005                 74 Euerka St. San Bernardino, Ca. 92408
2006                 123 4th St. San Bernardino, CA. 92407
2007                 14 Market Ave. San Bernardino, CA. 92408
2008                 35 Kinsey Ave. San Bernardino, CA. 92407
2009                 280 Drake Dr. San Bernardino, CA. 92408
2010                 654 Lincoln Ave. San Bernardino, Ca. 92408
2011                 719 Waterman Ave. San Bernardino, CA. 92408
2012                 4564 Waver Ave. San Bernardino, Ca. 92408
```

Rented_Residence - Continued

Data Loading: Normal Insert

Description: We used the normal insert method for this table. The insert script is below.

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
Data Load Method 1: Normal Insert - Insert SQL Command
Done By: Mark Takahashi
*/

INSERT INTO  rented_residence(residence_number,address)
values ('2001','4135 Sierra St. San Bernardino, CA. 92408');

INSERT INTO  rented_residence(residence_number,address)
values ('2002','1621 Euerka St. San Bernardino, CA. 92407');

INSERT INTO  rented_residence(residence_number,address)
values ('2003','1317 5th St. San Bernardino, CA. 92408');

INSERT INTO  rented_residence(residence_number,address)
values ('2004','14 University Ave, San Bernardino, CA. 92407');

INSERT INTO  rented_residence(residence_number,address)
values ('2005','74 Euerka St. San Bernardino, Ca. 92408');

INSERT INTO  rented_residence(residence_number,address)
values ('2006','123 4th St. San Bernardino, CA. 92407');

INSERT INTO  rented_residence(residence_number,address)
values ('2007','14 Market Ave. San Bernardino, CA. 92408');

INSERT INTO  rented_residence(residence_number,address)
values ('2008','35 Kinsey Ave. San Bernardino, CA. 92407');

INSERT INTO  rented_residence(residence_number,address)
values ('2009','280 Drake Dr. San Bernardino, CA. 92408');

INSERT INTO  rented_residence(residence_number,address)
values ('2010','654 Lincoln Ave. San Bernardino, Ca. 92408');

INSERT INTO  rented_residence(residence_number,address)
values ('2011','719 Waterman Ave. San Bernardino, CA. 92408');

INSERT INTO  rented_residence(residence_number,address)
values ('2012','4564 Waver Ave. San Bernardino, Ca. 92408');
```

Room

Team Member Responsible: Mark Takahashi

Table Comments

```
TABLE_NAME           TABLE_TYPE COMMENTS
-------------------- ---------- --------------------------------------------------
ROOM                 TABLE      A Room is where a student sleeps in either a Resid
                                ence Hall or Student Apartment, and is rented to t
                                he student
```

Column Comments

```
TABLE_NAME           COLUMN_NAME                    COMMENTS
-------------------- ------------------------------ --------------------------------------------------
ROOM                 ROOM_NUMBER                    The unique number associated with a Room - format:
                                                     1234

                     RENT                           The ammount paid to live in the Room - format: 100
                                                    .00

                     RESIDENCE_NUMBER               Foreign key referencing Residence Number of Rented
                                                     Residence - format: 1234
```

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Mark Takahashi
*/

CREATE TABLE Room (
Room_Number VARCHAR2(4)
      CONSTRAINT Room_Room_Number_UK UNIQUE,
Rent NUMBER(6,2)
      CONSTRAINT Room_Rent_NN NOT NULL,
Residence_Number CHAR(4),
CONSTRAINT Room_PK
      PRIMARY KEY (Room_Number, Residence_Number),
CONSTRAINT Room_Residence_Number_FK
      FOREIGN KEY (Residence_Number)
      REFERENCES Rented_Residence (Residence_Number) ON DELETE CASCADE
)
```

Room - Continued

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Table Structure

```
SQL> desc room
 Name                      Null?    Type
 ------------------------- -------- ----------------
 ROOM_NUMBER               NOT NULL VARCHAR2(4)
 RENT                      NOT NULL NUMBER(6,2)
 RESIDENCE_NUMBER          NOT NULL CHAR(4)
```

Table Contents

```
SQL> select *
  2  from room;

ROOM_NUMBER            RENT RESIDENCE_NUMBER
---------------- ---------- ------------------
3001                    300 2001
3002                    400 2002
3003                    500 2003
3004                    475 2004
3005                    480 2005
3006                    350 2006
3007                    600 2007
3008                    500 2008
3009                    400 2009
3010                    300 2010
3011                    350 2011
3012                    450 2012
```

Room - Continued

Data Loading: Sql Loader

Description: We used Sql Loader for this table. The ctl and dat files used are below

```
load data
append
into table Room
fields terminated by "/"
(room_number,rent,residence_number)
```

```
3001/300/2001
3002/400/2002
3003/500/2003
3004/475/2004
3005/480/2005
3006/350/2006
3007/600/2007
3008/500/2008
3009/400/2009
3010/300/2010
3011/350/2011
3012/450/2012
```

Residence_Hall

Team Member Responsible: Mark Takahashi

Table Comments

```
TABLE_NAME           TABLE_TYPE COMMENTS
-------------------- ---------- --------------------------------------------------
RESIDENCE_HALL       TABLE      A Residence Hall is a dormitory where students ren
                                t a room while taking courses
```

Column Comments

TABLE_NAME	COLUMN_NAME	COMMENTS
RESIDENCE_HALL	RH_NAME	The Name of the Residence Hall - format: Buckingham Palace Hall
	HALL_MANAGER	Name of person responsible for well being of Residence Hall - format: Joe Schmoe
	PHONE_NUMBER	The Phone Number of the Residence Hall - format: 9095551234
	RESIDENCE_NUMBER	Foreign key referencing Residence Number of Rented Residence - format: 1234

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Mark Takahashi
*/

CREATE TABLE Residence_Hall (
RH_Name VARCHAR2(25)
      CONSTRAINT RH_PK PRIMARY KEY,
Hall_Manager VARCHAR2(100)
      CONSTRAINT RH_Manager_NN NOT NULL,
Phone_Number VARCHAR2(10)
      CONSTRAINT RH_Phone_Number_NN NOT NULL,
Residence_Number CHAR(4)
      CONSTRAINT RH_Residence_Number_NN NOT NULL,
CONSTRAINT RH_Residence_Number_FK
      FOREIGN KEY (Residence_Number)
      REFERENCES Rented_Residence (Residence_Number)
)
```

Residence_Hall - Continued

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Table Structure

```
SQL> desc residence_hall
 Name                      Null?    Type
 ------------------------- -------- ----------------
 RH_NAME                   NOT NULL VARCHAR2(25)
 HALL_MANAGER              NOT NULL VARCHAR2(100)
 PHONE_NUMBER              NOT NULL VARCHAR2(10)
 RESIDENCE_NUMBER          NOT NULL CHAR(4)
```

Table Contents

```
SQL> select *
  2  from residence_hall;

RH_NAME                   HALL_MANAGER          PHONE_NUMBER   RESIDENCE_NUMBER
------------------------- --------------------- -------------- ----------------
Serrano Village           Jermey Piven          6547891451     2001
Arrowhead Village         Anthony Cooper        1234679998     2002
University Village        Melissa Anderson      6066061244     2003
Mountainside Village      Jacob Wright          1246571164     2004
Athena Village            Steven Ortega         9956544459     2005
Sunrise Village           Lisa Riton            6524567778     2006
```

Data Loading: Substitution

Description: We used the substitution method for this table. The substitution script is below.

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
Data Load Method 2: Insert Using Substitution Variables
Done By: Mark Takahashi
*/

INSERT INTO  residence_hall(RH_Name, Hall_Manager, Phone_Number, Residence_Number)
values ('&RH_Name','&Hall_Manager', '&Phone_Number', '&Residence_Number');
```

Student_Apartment

Team Member Responsible: James Small

Table Comments

```
TABLE_NAME           TABLE_TYPE COMMENTS
-------------------- ---------- --------------------------------------------------
STUDENT_APARTMENT    TABLE      This is the physical building that a student rents
                                 and resides in.
```

Column Comments

TABLE_NAME	COLUMN_NAME	COMMENTS
STUDENT_APARTMENT	APARTMENT_NUMBER	The number that uniquely identifies each apartment in the residence office - Format: 7001
	AVAILABLE_ROOMS	The number of rooms available in a particular student apartment - Format: 3
	STAFF_NUMBER	Foreign key referencing staff_number in staff_member. This identifies the staff_member who inspects a stduent_apartment periodically - Format: 1001
	RESIDENCE_NUMBER	Foreign key referencing residence_number in Rented_residence. This identifies the unique residence number among all residences managed by the residence office - Format: 2007

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: James Small
*/

CREATE TABLE Student_Apartment (
Apartment_Number VARCHAR2(6)
      CONSTRAINT SA_PK PRIMARY KEY,
Available_Rooms NUMBER(4)
      CONSTRAINT SA_Available_Rooms_NN NOT NULL,
Staff_Number CHAR(4) /*FK*/
      CONSTRAINT SA_Staff_Number_NN NOT NULL,
Residence_Number CHAR(4) /*FK*/
      CONSTRAINT SA_Residence_Number_NN NOT NULL,
CONSTRAINT SA_Staff_Number_FK
      FOREIGN KEY (Staff_Number)
      REFERENCES Staff_Member(Staff_Number) ON DELETE SET NULL,
CONSTRAINT SA_Residence_Number_FK
      FOREIGN KEY (Residence_Number)
      REFERENCES Rented_Residence(Residence_Number)
)
```

Student_Apartment - Continued

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Table Structure

```
SQL> desc student_apartment
 Name                      Null?    Type
 ------------------------- -------- ----------------
 APARTMENT_NUMBER          NOT NULL VARCHAR2(6)
 AVAILABLE_ROOMS           NOT NULL NUMBER(4)
 STAFF_NUMBER              NOT NULL CHAR(4)
 RESIDENCE_NUMBER          NOT NULL CHAR(4)
```

Table Contents

```
SQL> select *
  2  from student_apartment;

APARTMENT_NUMBER     AVAILABLE_ROOMS STAFF_NUMBER    RESIDENCE_NUMBER
-------------------- --------------- --------------- --------------------
7001                               3 1001            2007
7002                               4 1002            2008
7003                               1 1003            2009
7004                               0 1003            2010
7005                               2 1002            2011
7006                               1 1001            2012
```

Data Loading: Sql Loader

Description: We used Sql Loader for this table. The ctl and dat files used are below

```
load data
append
into table student_apartment
fields terminated by "/"
(apartment_number,available_rooms,staff_number,residence_number)
```

Student_Apartment - Continued

```
7001/3/1001/2007
7002/4/1002/2008
7003/1/1003/2009
7004/0/1003/2010
7005/2/1002/2011
7006/1/1001/2012
```

Inspection_Report

Team Member Responsible: Daniel Urbach

Table Comments

TABLE_NAME	TABLE_TYPE	COMMENTS
INSPECTION_REPORT	TABLE	Inspection_Report is a summary of an inspection done on a student apartment by a staff member

Column Comments

TABLE_NAME	COLUMN_NAME	COMMENTS
INSPECTION_REPORT	INSPECTION_DATE	The date and time when an Inspection was conducted - format: 01-MAY-11 12:00:00
	SATISFACTORY	The overall status of the Inspection Report - format: Unsatisfactory
	COMMENTS	Additional comments on the status of Student Apartment, made by the Staff Member who makes the Inspection Report - format: The students in this apartment are super cool, and they keep a clean shop. However one of the chairs at their diningroom table is broken; I questioned them about this and they claimed it was that way when they moved in.
	APARTMENT_NUMBER	Foreign key referencing Apartment_Number in Student Apartment, the unique identifier for Student Apartment - format: 7001

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Daniel Urbach
*/

CREATE TABLE Inspection_Report (
Inspection_Date DATE,
Satisfactory CHAR(20)
     CONSTRAINT IR_Satisfactory_NN NOT NULL,
Comments VARCHAR2(4000)
     CONSTRAINT IR_Comments_NN NOT NULL,
Apartment_Number VARCHAR2(6),
CONSTRAINT Inspection_Report_PK
     PRIMARY KEY (Inspection_Date, Apartment_Number),
CONSTRAINT IR_Apartment_Number_FK
     FOREIGN KEY (Apartment_Number)
     REFERENCES Student_Apartment (Apartment_Number) ON DELETE CASCADE
)
```

Inspection_Report - Continued

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Table Structure

```
SQL> desc inspection_report
 Name                      Null?    Type
 ------------------------- -------- ----------------
 INSPECTION_DATE           NOT NULL DATE
 SATISFACTORY              NOT NULL CHAR(20)
 COMMENTS                  NOT NULL VARCHAR2(4000)
 APARTMENT_NUMBER          NOT NULL VARCHAR2(6)
```

Table Contents

```
SQL> select *
  2  from inspection_report;

INSPECTION_DATE      SATISFACTORY         COMMENTS                                                        APARTMENT_NUMBER
-------------------- -------------------- --------------------------------------------------------------- --------------------
15-OCT-12            Satisfactory         Apartment in satisfactory condition.  No problems               7001
16-OCT-12            Unsatisfactory       Apartment in unsatisfactory condition.  Holes in walls.        7002
17-OCT-12            Satisfactory         Apartment in satisfactory condition.  No problems               7003
18-OCT-12            Unsatisfactory       Apartment in unsatisfactory condition.  Carpet had fire damage  7004
19-OCT-12            Unsatisfactory       Apartment in unsatisfactory condition.  Spray Paint on walls    7005
20-OCT-12            Satisfactory         Apartment in satisfactory condition.  No problems               7006
15-FEB-13            Satisfactory         Apartment in satisfactory condition.  No problems               7001
16-FEB-13            Satisfactory         Apartment in satisfactory condition.  No problems               7002
```

Data Loading: Sql Loader

Description: We used Sql Loader for this table. The ctl and dat files used are below

```
load data
append
into table inspection_report
fields terminated by "/"
(inspection_date,satisfactory,comments,apartment_number)
```

```
15-OCT-12/Satisfactory/Apartment in satisfactory condition.  No problems/7001
16-OCT-12/Unsatisfactory/Apartment in unsatisfactory condition.  Holes in walls./7002
17-OCT-12/Satisfactory/Apartment in satisfactory condition.  No problems/7003
18-OCT-12/Unsatisfactory/Apartment in unsatisfactory condition.  Carpet had fire damage/7004
19-OCT-12/Unsatisfactory/Apartment in unsatisfactory condition.  Spray Paint on walls/7005
20-OCT-12/Satisfactory/Apartment in satisfactory condition.  No problems/7006
15-FEB-13/Satisfactory/Apartment in satisfactory condition.  No problems/7001
16-FEB-13/Satisfactory/Apartment in satisfactory condition.  No problems/7002
```

Rents

Team Member Responsible: Mark Takahashi

Table Comments

```
TABLE_NAME           TABLE_TYPE COMMENTS
-------------------- ---------- --------------------------------------------------
RENTS                TABLE      Rents represents the ternary relationship between
                                the Student, Room and Lease
```

Column Comments

TABLE_NAME	COLUMN_NAME	COMMENTS
RENTS	PLACE_NUMBER	The unique identifier of each instance of a Student renting a room - format: 1234
	STUDENT_NUMBER	Foreign key referencing the Student Number of Student who is renting a room - format: 123456
	ROOM_NUMBER	Foreign key referencing the Room Number of the Room being rented - format: 1234
	LEASE_NUMBER	Foreign key referencing the Lease Number of the Lease contracting the room rental to the student - format: 12345

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Mark Takahashi
*/

CREATE TABLE Rents (
Place_Number VARCHAR2(4)
      CONSTRAINT Rents_Place_Number_NN NOT NULL,
Student_Number VARCHAR2(6),
Room_Number VARCHAR2(4),
Lease_Number VARCHAR2(5),
CONSTRAINT Rents_PK
      PRIMARY KEY (Student_Number, Room_Number, Lease_Number),
CONSTRAINT Rents_Student_Number_FK
      FOREIGN KEY (Student_Number)
      REFERENCES Student (Student_Number) ON DELETE CASCADE,
CONSTRAINT Rents_Room_Number_FK
      FOREIGN KEY (Room_Number)
      REFERENCES Room (Room_Number) ON DELETE CASCADE,
CONSTRAINT Rents_Lease_Number_FK
      FOREIGN KEY (Lease_Number)
      REFERENCES Lease (Lease_Number) ON DELETE CASCADE
)
```

Rents - Continued

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Table Structure

```
SQL> desc rents
 Name                      Null?    Type
 ------------------------- -------- ----------------
 PLACE_NUMBER              NOT NULL VARCHAR2(4)
 STUDENT_NUMBER            NOT NULL VARCHAR2(6)
 ROOM_NUMBER               NOT NULL VARCHAR2(4)
 LEASE_NUMBER              NOT NULL VARCHAR2(5)
```

Table Contents

```
SQL> select *
  2  from rents;

PLACE_NUMBER    STUDENT_NUMBER  ROOM_NUMBER      LEASE_NUMBER
--------------- --------------- ---------------- ---------------
1               600014          3001             10000
2               600002          3002             10001
3               600001          3003             10002
4               600003          3004             10003
5               600004          3005             10004
6               600007          3006             10005
7               600005          3007             10006
8               600009          3008             10007
9               600006          3009             10008
10              600008          3010             10009
11              600010          3011             10010
12              600011          3012             10011
```

Rents - Continued

Data Loading: Sql Loader

Description: We used Sql Loader for this table. The ctl and dat files used are below

```
load data
append
into table rents
fields terminated by "/"
(place_number,student_number,room_number,lease_number)
```

```
1/600014/3001/10000
2/600002/3002/10001
3/600001/3003/10002
4/600003/3004/10003
5/600004/3005/10004
6/600007/3006/10005
7/600005/3007/10006
8/600009/3008/10007
9/600006/3009/10008
10/600008/3010/10009
11/600010/3011/10010
12/600011/3012/10011
```

Invoice

Team Member Responsible: Enrique Carbajal

Table Comments

TABLE_NAME	TABLE_TYPE	COMMENTS
INVOICE	TABLE	An invoice is a receipt given to student as a summary of lease terms, monies paid, and other pertinent information.

Column Comments

TABLE_NAME	COLUMN_NAME	COMMENTS
INVOICE	INVOICE_NUMBER	Primary key identifying the invoice -- Format: 4001
	PAYMENT_METHOD	The method of payment the student presented --Format: Credit
	RENTAL_ADDRESS	The address of the rental property -- Format: 1317 5th St. San Bernardino, CA. 92408
	PAYMENT_RECEIVED_DATE	The date of when the payment was received-- Format:9/25/2012
	ROOM_NUMBER	The room number that was leased-- Format:9/25/2012
	PAYMENT_DUE_DATE	The due date of the payment -- Format: 9/30/2012
	PLACE_NUMBER	The place number the invoice refers to -- Format: 3
	STUDENT_NUMBER	Foreign Key referencing Student relation. The student number that is the owner of the invoice -- Format: 600014
	LEASE_NUMBER	Foreign key referencing Lease Number table. This is the Lease Number that this invoice belongs to -- Format:10000

Invoice - Continued

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Enrique Carbajal
*/

CREATE TABLE Invoice (
Invoice_Number VARCHAR2(4)
      CONSTRAINT Invoice_PK PRIMARY KEY,
Payment_Method VARCHAR2(6)
      CONSTRAINT Invoice_Payment_Method_NN NOT NULL,
Rental_Address VARCHAR2(300)
      CONSTRAINT Invoice_Rental_Address_NN NOT NULL,
Payment_Received_Date DATE,
Room_Number VARCHAR2(7)
      CONSTRAINT Invoice_Room_Number_NN NOT NULL,
Payment_Due_Date DATE
      CONSTRAINT Invoice_Payment_Due_Date_NN NOT NULL,
Place_Number VARCHAR2(4)
      CONSTRAINT Invoice_Place_Number_NN NOT NULL,
Student_Number VARCHAR2(6)
      CONSTRAINT Invoice_Student_Number_NN NOT NULL,
Lease_Number VARCHAR2(5)
      CONSTRAINT Invoice_Lease_Number_NN NOT NULL,
CONSTRAINT Invoice_Student_Number_FK
      FOREIGN KEY (Student_Number)
      REFERENCES Student (Student_Number) ON DELETE CASCADE,
CONSTRAINT Invoice_Lease_Number_FK
      FOREIGN KEY (Lease_Number)
      REFERENCES Lease (Lease_Number) ON DELETE CASCADE
)
```

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Invoice - Continued

Table Structure

```
SQL> desc invoice
 Name                          Null?    Type
 ----------------------------- -------- ----------------
 INVOICE_NUMBER                NOT NULL VARCHAR2(4)
 PAYMENT_METHOD                NOT NULL VARCHAR2(6)
 RENTAL_ADDRESS                NOT NULL VARCHAR2(300)
 PAYMENT_RECEIVED_DATE                  DATE
 ROOM_NUMBER                   NOT NULL VARCHAR2(7)
 PAYMENT_DUE_DATE              NOT NULL DATE
 PLACE_NUMBER                  NOT NULL VARCHAR2(4)
 STUDENT_NUMBER                NOT NULL VARCHAR2(6)
 LEASE_NUMBER                  NOT NULL VARCHAR2(5)
```

Table Contents

```
SQL> select *
  2  from invoice;

INVOICE_NUMBER PAYMENT_METHOD RENTAL_ADDRESS                                PAYMENT_RECEIVED_DATE     ROOM_NUMBER PAYMENT_DUE_DATE  PLACE_NUMBER STUDENT_NUMBER LEASE_NUMBER
-------------- -------------- ------------------------------------------- ------------------------- ----------- ----------------- ------------ -------------- ------------
4001           Credit         1317 5th St. San Bernardino, CA. 92408      25-SEP-12                 3003        30-SEP-12         3            600001         10002
4002           Debit          1621 Euerka St. San Bernardino, CA. 92407   25-SEP-12                 3002        30-SEP-12         2            600002         10001
4003           Cash           14 University Ave, San Bernardino, CA. 92407 14-SEP-12                3004        30-SEP-12         4            600003         10003
4004           Credit         74 Euerka St. San Bernardino, Ca. 92408     26-SEP-12                 3005        30-SEP-12         5            600004         10004
4005           Cash           14 Market Ave. San Bernardino, CA. 92408    24-SEP-12                 3007        30-SEP-12         7            600005         10006
4006           Credit         123 4th St. San Bernardino, CA. 92407       03-OCT-12                 3006        30-SEP-12         6            600007         10005
4007           Check          280 Drake Dr. San Bernardino, CA. 92408     29-SEP-12                 3009        30-SEP-12         9            600006         10008
4008           Credit         654 Lincoln Ave. San Bernardino, Ca. 92408  27-SEP-12                 3010        30-SEP-12         10           600008         10009
4009           Check          35 Kinsey Ave. San Bernardino, CA. 92407    29-SEP-12                 3008        30-SEP-12         8            600009         10007
4010           Check          719 Waterman Ave. San Bernardino, CA. 92408 28-SEP-12                 3011        30-SEP-12         11           600010         10010
4011           Check          35 Kinsey Ave. San Bernardino, CA. 92407    14-JAN-13                 3008        30-JAN-13         8            600009         10007
4012           Credit         4564 Waver Ave. San Bernardino, Ca. 92408   24-SEP-12                 3012        30-SEP-12         12           600011         10011
4013           Check          719 Waterman Ave. San Bernardino, CA. 92408 13-JAN-13                 3011        30-JAN-13         11           600010         10010
4014           Check          4135 Sierra St. San Bernardino, CA. 92408   27-SEP-12                 3001        30-SEP-12         1            600014         10000
4015           Debit          1621 Euerka St. San Bernardino, CA. 92407   10-FEB-13                 3002        30-JAN-13         2            600002         10001
4016           Credit         1317 5th St. San Bernardino, CA. 92408      27-JAN-13                 3003        30-JAN-13         3            600001         10002
4017           Cash           14 University Ave, San Bernardino, CA. 92407 27-JAN-13                3004        30-JAN-13         4            600003         10003
4018           Cash           14 Market Ave. San Bernardino, CA. 92408    27-JAN-13                 3007        30-JAN-13         7            600005         10006
```

Invoice - Continued

Data Loading: Normal Insert

Description: We used the normal insert method for this table. The insert script is below.

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
Data Load Method 1: Normal Insert - Insert SQL Command
Done By: Enrique Carbajal Jr.
*/

INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ( '4001', 'Credit', '1317 5th St. San Bernardino, CA. 92408', '25-SEP-12', '3003', '30-SEP-12',     '3', '600001', '10002');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ( '4002', 'Debit', '1621 Euerka St. San Bernardino, CA. 92407', '25-SEP-12', '3002', '30-SEP-12', '2', '600002', '10001');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ( '4003','Cash',    '14 University Ave, San Bernardino, CA. 92407', '14-SEP-12','3004','30-SEP-12', '4', '600003','10003');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ( '4004','Credit','74 Euerka St. San Bernardino, Ca. 92408','26-SEP-12','3005','30-SEP-12','5','600004','10004');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ( '4005','Cash','14 Market Ave. San Bernardino, CA. 92408','24-SEP-12','3007','30-SEP-12','7','600005','10006');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ( '4006','Credit','123 4th St. San Bernardino, CA. 92407','03-OCT-12','3006','30-SEP-12','6','600007','10005');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ( '4007','Check','280 Drake Dr. San Bernardino, CA. 92408','29-SEP-12','3009','30-SEP-12','9','600006','10008');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ('4008','Credit','654 Lincoln Ave. San Bernardino, Ca. 92408','27-SEP-12','3010','30-SEP-12','10','600008','10009');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ('4009','Check','35 Kinsey Ave. San Bernardino, CA. 92407','29-SEP-12','3008','30-SEP-12','8','600009','10007');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ('4010','Check','719 Waterman Ave. San Bernardino, CA. 92408','28-SEP-12','3011','30-SEP-12','11','600010','10010');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ('4011','Check','35 Kinsey Ave. San Bernardino, CA. 92407','14-JAN-13','3008','30-JAN-13','8','600009','10007');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ('4012','Credit','4564 Waver Ave. San Bernardino, Ca. 92408','24-SEP-12','3012','30-SEP-12','12','600011','10011');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ('4013','Check','719 Waterman Ave. San Bernardino, CA. 92408','13-JAN-13','3011','30-JAN-13','11','600010','10010');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ('4014','Check','4135 Sierra St. San Bernardino, CA. 92408','27-SEP-12','3001','30-SEP-12','1','600014', '10000');


INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ('4015','Debit','1621 Euerka St. San Bernardino, CA. 92407','10-FEB-13','3002','30-JAN-13','2','600002','10001');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ('4016','Credit','1317 5th St. San Bernardino, CA. 92408','27-JAN-13','3003','30-JAN-13','3','600001','10002');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ('4017','Cash','14 University Ave, San Bernardino, CA. 92407','27-JAN-13','3004','30-JAN-13','4','600003','10003');
INSERT INTO Invoice (Invoice_Number, Payment_Method, Rental_Address, Payment_Received_Date, Room_Number, Payment_Due_Date, Place_Number, Student_Number,
Lease_Number)
values ('4018','Cash','14 Market Ave. San Bernardino, CA. 92408','27-JAN-13','3007','30-JAN-13','7','600005','10006');
```

Reminder

Team Member Responsible: Enrique Carbajal

Table Comments

```
TABLE_NAME           TABLE_TYPE COMMENTS
-------------------- ---------- --------------------------------------------------
REMINDER             TABLE      This table is used to track the date a reminder wa
                                s sent pertaining to a particular invoice number
```

Column Comments

```
TABLE_NAME           COLUMN_NAME                    COMMENTS
-------------------- ------------------------------ --------------------------------------------------
REMINDER             REMINDER_DATE                  This is the date reminding the student of when pay
                                                    ment is due -- Format: 9/25/2012

                     INVOICE_NUMBER                 This is a foreign key refrencing invoice_number in
                                                     table Invoice.  -- Format: 4001
```

Create Table

```
/*
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
CREATED BY: Enrique Carbajal
*/

CREATE TABLE Reminder (
Reminder_Date DATE,
Invoice_Number VARCHAR2(4)
     CONSTRAINT Reminder_PK PRIMARY KEY (Reminder_Date, Invoice_Number),
CONSTRAINT Reminder_Invoice_Number_FK FOREIGN KEY (Invoice_Number) REFERENCES Invoice (Invoice_Number) ON DELETE CASCADE
)
```

Note: No alter table commands were needed. All constraints were added on table creation due to the order that we created our tables. The order of the tables in this document was thought out ahead of time so that we wouldn't need to use alter table commands to add constraints. The data insertion also followed this same order.

Table Structure

```
SQL> desc reminder
 Name                      Null?    Type
 ------------------------- -------- ----------------
 REMINDER_DATE             NOT NULL DATE
 INVOICE_NUMBER            NOT NULL VARCHAR2(4)
```

Reminder - Continued

Table Contents

```
SQL> select *
  2  from reminder;

REMINDER_DATE      INVOICE_NUMBER
------------------ ---------------
15-SEP-12          4001
15-SEP-12          4002
15-SEP-12          4004
15-SEP-12          4005
15-SEP-12          4006
15-SEP-12          4007
15-SEP-12          4008
15-SEP-12          4009
15-SEP-12          4010
15-SEP-12          4012
15-SEP-12          4014
01-OCT-12          4006
15-JAN-13          4015
15-JAN-13          4016
15-JAN-13          4017
15-JAN-13          4018
01-FEB-13          4015
```

Data Loading: Sql Loader

Description: We used Sql Loader for this table. The ctl and dat files used are below

```
load
append
into table Reminder
fields terminated by "/"
(Reminder_Date, Invoice_Number)
```

```
15-SEP-12/4001
15-SEP-12/4002
15-SEP-12/4004
15-SEP-12/4005
15-SEP-12/4006
01-OCT-12/4006
15-SEP-12/4007
15-SEP-12/4008
15-SEP-12/4009
15-SEP-12/4010
15-SEP-12/4012
15-SEP-12/4014
15-JAN-13/4015
01-FEB-13/4015
15-JAN-13/4016
15-JAN-13/4017
15-JAN-13/4018
```

Project Query 1

Team Member Responsible: Mark Takahashi

Description

1. Present a report listing the Manager's name and telephone number for each hall of residence.

SQL

```
/*
Present a report listing the Manager's name and telephone number for each hall of residence.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: Mark Takahashi
*/

set linesize 400
SET FEEDBACK OFF
TTITLE "Query 1"
COLUMN hall_manager HEADING "Hall Manager"
COLUMN phone_number HEADING "Phone Number"
COLUMN hall_manager format a20
COLUMN phone_number format a15

select  Hall_Manager, Phone_Number
from Residence_Hall;

SET FEEDBACK ON
COLUMN hall_manager CLEAR
COLUMN phone_number CLEAR
```

Result

```
Hall Manager         Phone Number
-------------------- ---------------
Jermey Piven         6547891451
Anthony Cooper       1234679998
Melissa Anderson     6066061244
Jacob Wright         1246571164
Steven Ortega        9956544459
Lisa Riton           6524567778
```

Project Query 2

Team Member Responsible: Mark Takahashi

Description

2. Present a report listing the names and student ids of students with the details of their lease agreements.

SQL

```
/*
Present a report listing the names and student ids of students with the details of their lease agreements.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: Mark Takahashi
*/

set linesize 400
set pagesize 100
SET FEEDBACK OFF
TTITLE "Query 2"
COLUMN first_name HEADING "First Name" format a10
COLUMN last_name HEADING "Last Name" format a10
COLUMN STUDENT_NUMBER HEADING "Student ID" format a12
COLUMN lease_number HEADING "Lease #" format a12
COLUMN duration HEADING "Duration" format a10
COLUMN rental_address HEADING "Rental Address" format a45
COLUMN move_out_date HEADING "Move Out Date" format a15
COLUMN move_in_date HEADING "Move In Date" format a15
COLUMN room_number HEADING "Room #" format a12
COLUMN phone_number HEADING "Phone #" format a12

select s.first_name, s.last_name, s.student_number, l.lease_number, l.duration,
         l.rental_address, l.move_out_date, l.move_in_date, l.room_number, l.phone_number
from student s, lease l, rents r
where s.student_number = r.student_number AND r.lease_number = l.lease_number;

SET FEEDBACK ON
COLUMN first_name CLEAR
COLUMN last_name CLEAR
COLUMN student_number CLEAR
COLUMN lease_number CLEAR
COLUMN duration CLEAR
COLUMN rental_address CLEAR
COLUMN move_out_date CLEAR
COLUMN move_in_date ClEAR
COLUMN room_number CLEAR
COLUMN phone_number CLEAR
```

Result

First Name	Last Name	Student ID	Lease #	Duration	Rental Address	Move Out Date	Move In Date	Room #	Phone #
Curtis	Wright	600001	10002	4	1317 5th St. San Bernardino CA. 92408	15-AUG-12	30-AUG-12	3003	9091111111
Michael	Baldridge	600002	10001	3	1621 Euerka St. San Bernardino CA. 92407	15-JUN-12	30-AUG-12	3002	9092222222
Liz	Smith	600003	10003	2	14 University Ave. San Bernardino CA. 92407	15-MAR-12	30-AUG-12	3004	9093333333
Andrew	Sherwood	600004	10004	1	74 Euerka St. San Bernardino Ca. 92408	15-DEC-12	30-AUG-12	3005	9094444444
Jennifer	West	600007	10005	1	123 4th St. San Bernardino CA. 92407	15-DEC-12	30-AUG-12	3006	9097777777
Melanie	Williams	600014	10000	1	4135 Sierra St. San Bernardino CA. 92408	15-DEC-12	30-AUG-12	3001	9093033030
Melissa	Vaughn	600005	10006	4	14 Market Ave. San Bernardino CA. 92408	15-AUG-12	30-AUG-12	3007	9095555555
Beth	Lopez	600006	10008	3	280 Drake Dr. San Bernardino CA. 92408	15-JUN-12	30-AUG-12	3009	9096666666
John	Wiley	600008	10009	1	654 Lincoln Ave. San Bernardino Ca. 92408	15-DEC-12	30-AUG-12	3010	9098888888
Jessica	Wilson	600009	10007	4	35 Kinsey Ave. San Bernardino CA. 92407	15-AUG-12	30-AUG-12	3008	9099999999
Ryan	Oliver	600010	10010	2	719 Waterman Ave. San Bernardino CA. 92408	15-MAR-12	30-AUG-12	3011	9091234567
Jesse	Gomez	600011	10011	1	4564 Waver Ave. San Bernardino Ca. 92408	15-DEC-12	30-AUG-12	3012	9097654321

Project Query 3

Team Member Responsible: Mark Takahashi

Description

3. Display the details of lease agreements that include the summer quarter.

SQL

```
/*
Display the details of lease agreements that include the summer quarter.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: Mark Takahashi
*/

set linesize 400
set pagesize 100
SET FEEDBACK OFF
TTITLE "Query 3"
COLUMN lease_number HEADING "Lease #" format a12
COLUMN duration HEADING "Duration" format a10
COLUMN rental_address HEADING "Rental Address" format a45
COLUMN move_out_date HEADING "Move Out Date" format a15
COLUMN move_in_date HEADING "Move In Date" format a15
COLUMN room_number HEADING "Room #" format a7
COLUMN phone_number HEADING "Phone #" format a12

select lease_number, duration, rental_address, move_out_date, move_in_date, room_number, phone_number
from lease
where duration = '4';

SET FEEDBACK ON
COLUMN lease_number CLEAR
COLUMN duration CLEAR
COLUMN rental_address CLEAR
COLUMN move_out_date CLEAR
COLUMN move_in_date CLEAR
COLUMN room_number CLEAR
COLUMN phone_number CLEAR
TTITLE OFF
```

Result

```
Lease #      Duration   Rental Address                               Move Out Date   Move In Date    Room #  Phone #
------------ ---------- -------------------------------------------- --------------- --------------- ------- ------------
10002        4          1317 5th St. San Bernardino CA. 92408        15-AUG-12       30-AUG-12       3003    9091111111
10006        4          14 Market Ave. San Bernardino CA. 92408      15-AUG-12       30-AUG-12       3007    9095555555
10007        4          35 Kinsey Ave. San Bernardino CA. 92407      15-AUG-12       30-AUG-12       3008    9099999999
```

Project Query 4

Team Member Responsible: Enrique Carbajal

Description

4. Display the details of the total rent paid by a given student.

SQL

```
/*
Display the details of the total rent paid by a given student.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: Enrique Carbajal
*/

set linesize 400
set pagesize 100
SET FEEDBACK OFF
TTITLE "Query 4"
COLUMN first_name HEADING "First Name" format a15
COLUMN last_name HEADING "Last Name" format a15

select s.first_name, s.last_name, sum(r.rent) "Total Rent"
from student s, invoice i, rents rt, room r
where s.student_number = i.student_number AND i.payment_received_date < sysdate
        and s.student_number = rt.student_number and rt.room_number = r.room_number
group by s.first_name, s.last_name;

SET FEEDBACK ON
COLUMN first_name CLEAR
COLUMN last_name CLEAR
TTITLE OFF
```

Result

First Name	Last Name	Total Rent
Melissa	Vaughn	1200
Liz	Smith	950
Beth	Lopez	400
Jessica	Wilson	1000
Jesse	Gomez	450
Andrew	Sherwood	480
Curtis	Wright	1000
Melanie	Williams	300
John	Wiley	300
Ryan	Oliver	700
Michael	Baldridge	800
Jennifer	West	350

Project Query 5

Team Member Responsible: Enrique Carbajal

Description

5. Present a report on students who have not paid their invoices by a given date.

SQL

```
/*
Present a report on students who have not paid their invoices by a given date.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: Enrique Carbajal
*/

set linesize 400
set pagesize 100
TTITLE "Query 5"
SET FEEDBACK OFF
COLUMN first_name HEADING "First Name" format a15
COLUMN last_name HEADING "Last Name" format a15
COLUMN student_number HEADING "Student ID" format a12

select s.first_name, s.last_name, s.student_number
from student s, invoice i
where s.student_number = i.student_number AND i.payment_received_date > '&Date';

TTITLE OFF
SET FEEDBACK ON
COLUMN first_name CLEAR
COLUMN last_name CLEAR
COLUMN student_number CLEAR
```

Result

```
Enter value for date: 01-FEB-13
old   3: where s.student_number = i.student_number AND i.payment_received_date > '&Date'
new   3: where s.student_number = i.student_number AND i.payment_received_date > '01-FEB-13'

Thu Nov 28
      page    1


First Name      Last Name       Student ID
--------------- --------------- ------------
Michael         Baldridge       600002
```

Project Query 6

Team Member Responsible: Enrique Carbajal

Description

6. Display the details of apartment inspections where the property was found to be in an unsatisfactory condition.

SQL

```
/*
Display the details of apartment inspections where the property was
        found to be in an unsatisfactory condition.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: Enrique Carbajal
*/

set linesize 400
set pagesize 50
TTITLE "Query 6"
SET FEEDBACK OFF
COLUMN inspection_date HEADING "Inspection Date" format a17
COLUMN satisfactory HEADING "Satisfactory" format a17
COLUMN comments HEADING "Comments" format a62
COLUMN apartment_number HEADING "Apt #" format a10

select inspection_date, satisfactory, comments, apartment_number
from inspection_report
where satisfactory = 'Unsatisfactory';

TTITLE OFF
SET FEEDBACK ON
COLUMN inspection_date CLEAR
COLUMN satisfactory CLEAR
COLUMN comments CLEAR
COLUMN apartment_number CLEAR
```

Result

Inspection Date	Satisfactory	Comments	Apt #
16-OCT-12	Unsatisfactory	Apartment in unsatisfactory condition. Holes in walls.	7002
18-OCT-12	Unsatisfactory	Apartment in unsatisfactory condition. Carpet had fire damage	7004
19-OCT-12	Unsatisfactory	Apartment in unsatisfactory condition. Spray Paint on walls	7005

Project Query 7

Team Member Responsible: James Small

Description

7. Present a report of the names and student ids of students with their room number and place number in a particular hall of residence.

SQL

```
/*
Present a report of the names and student ids of students with their
        room number and place number in a particular hall of residence.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: James Small
*/

set linesize 400
set pagesize 100
TTITLE "Query 7"
SET FEEDBACK OFF
COLUMN first_name HEADING "First Name" format a15
COLUMN last_name HEADING "Last Name" format a15
COLUMN student_number HEADING "Student ID" format a12
COLUMN room_number HEADING "Room #" format a12
COLUMN place_number HEADING "Place #" format a12

select s.first_name, s.last_name, s.student_number, r.room_number, r.place_number
from student s, rents r
where s.student_number = r.student_number AND s.residence_type = 'Residence Hall';

SET FEEDBACK ON
TTITLE OFF
COLUMN first_name CLEAR
COLUMN last_name CLEAR
COLUMN student_number CLEAR
COLUMN room_number CLEAR
COLUMN place_number CLEAR
```

Result

First Name	Last Name	Student ID	Room #	Place #
Curtis	Wright	600001	3003	3
Michael	Baldridge	600002	3002	2
Liz	Smith	600003	3004	4
Andrew	Sherwood	600004	3005	5
Jennifer	West	600007	3006	6
Melanie	Williams	600014	3001	1

Project Query 8

Team Member Responsible: James Small

Description

8. Present a report listing the details of all students currently on the waiting list for accommodation; that is, who were not placed.

SQL

```
/*
Present a report listing the details of all students currently on the waiting
        list for accommodation; that is, who were not placed.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: James Small
*/

set linesize 400
set pagesize 100
TTITLE "Query 8"
SET FEEDBACK OFF
COLUMN first_name HEADING "First Name" format a15
COLUMN last_name HEADING "Last Name" format a15
COLUMN student_number HEADING "Student ID" format a12

select first_name, last_name, student_number
from student
where current_status = 'Waiting';

SET FEEDBACK ON
TTITLE OFF
COLUMN first_name CLEAR
COLUMN last_name CLEAR
COLUMN student_number CLEAR
```

Result

```
First Name      Last Name       Student ID
--------------- --------------- ------------
Paul            Darkwa          600012
Susan           Storey          600013
```

Project Query 9

Team Member Responsible: Daniel Urbach

Description

9. Display the total number of students in each student category.

SQL

```
/*
Display the total number of students in each student category.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: Daniel Urbach
*/

set linesize 400
set pagesize 100
SET FEEDBACK OFF
TTITLE "Query 9"
COLUMN class HEADING "Class" format a15

select class, count(*) "Count"
from student
group by class;

SET FEEDBACK ON
TTITLE OFF
COLUMN class CLEAR
```

Result

```
Class                Count
---------------  ----------
Sophomore                3
Senior                   3
Freshman                 5
Junior                   3
```

Project Query 10

Team Member Responsible: Daniel Urbach

Description

10. Present a report of the names and ids for all students who have not supplied details of their next-of-kin.

SQL

```
/*
Present a report of the names and ids for all students who have
        not supplied details of their next-of-kin.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: Daniel Urbach
*/

set linesize 400
set pagesize 100
SET FEEDBACK OFF
TTITLE "Query 10"
COLUMN first_name HEADING "First Name" format a15
COLUMN last_name HEADING "Last Name" format a15
COLUMN student_number HEADING "Student ID" format a12

select s.first_name, s.last_name, s.student_number
from student s
where not exists (select *
                  from nok_relationship r
                  where s.student_number = r.student_number);

SET FEEDBACK ON
TTITLE OFF
COLUMN first_name CLEAR
COLUMN last_name CLEAR
COLUMN student_number CLEAR
```

Result

```
First Name      Last Name       Student ID
--------------- --------------- ------------
Liz             Smith           600003
Melissa         Vaughn          600005
Jennifer        West            600007
Jessica         Wilson          600009
Jesse           Gomez           600011
```

Project Query 11

Team Member Responsible: Daniel Urbach

Description

11.Display the name and internal telephone number of the Adviser for a particular student.

SQL

```
/*
Display the name and internal telephone number of the
        Adviser for a particular student.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: Daniel Urbach
*/

set linesize 400
set pagesize 100
SET FEEDBACK OFF
TTITLE "Query 11"
COLUMN first_name HEADING "First Name" format a15
COLUMN last_name HEADING "Last Name" format a15
COLUMN phone_number HEADING "Phone #" format a12

select m.first_name, m.last_name, a.phone_number
from student s, staff_member m, advisor a
where s.student_number = '&Student_ID' AND m.staff_number = a.advisor_number AND s.advisor_number = a.advisor_number;

SET FEEDBACK ON
TTITLE OFF
COLUMN first_name CLEAR
COLUMN last_name CLEAR
COLUMN phone_number CLEAR
```

Result

```
Enter value for student_id: 600001
old   3: where s.student_number = '&Student_ID' AND m.staff_number = a.advisor_number AND s.advisor_number = a.advisor_number
new   3: where s.student_number = '600001' AND m.staff_number = a.advisor_number AND s.advisor_number = a.advisor_number

Thu Nov 28
      page    1


First Name      Last Name       Phone #
--------------- --------------- ------------
Michelle        Smith           9095551234
```

Project Query 12

Team Member Responsible: James Small

Description

12.Display the minimum, maximum, and average monthly rent for rooms in residence halls.

SQL

```
/*
Display the minimum, maximum, and average monthly rent for rooms in residence halls.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: James Small
*/

set linesize 400
set pagesize 100
SET FEEDBACK OFF
TTITLE "Query 12"

select min(ro.rent) "Min Rent", max(ro.rent) "Max Rent", avg(ro.rent) "Avg Rent"
from room ro,rented_residence rr, residence_hall rh
where ro.residence_number = rr.residence_number AND rr.residence_number = rh.residence_number;

SET FEEDBACK ON
TTITLE OFF
```

Result

```
  Min Rent   Max Rent   Avg Rent
---------- ---------- ----------
       300        500      417.5
```

Project Query 13

Team Member Responsible: James Small

Description

13.Display the staff number, name, age, and current location of all members of the residence staff who are over 60 years of age today.

SQL

```
/*
Display the staff number, name, age, and current location of all members of
        the residence staff who are over 60 years of age today.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: James Small
*/

set linesize 400
set pagesize 100
SET FEEDBACK OFF
TTITLE "Query 13"
COLUMN staff_number HEADING "Staff #" format a8
COLUMN first_name HEADING "First Name" format a15
COLUMN last_name HEADING "Last Name" format a15
COLUMN location HEADING "Location" format a15

select staff_number, first_name, last_name,
        to_number(to_char(sysdate,'YYYY'))-to_number(to_char(19)|| to_char(date_of_birth,'YY')) "Age", location
from staff_member
where to_number(to_char(sysdate,'YYYY'))-to_number(to_char(19)|| to_char(date_of_birth,'YY')) >= 60;

SET FEEDBACK ON
TTITLE OFF
COLUMN staff_number CLEAR
COLUMN first_name CLEAR
COLUMN last_name CLEAR
COLUMN location CLEAR
```

Result

Staff #	First Name	Last Name	Age	Location
1000	Anthony	Gonzalez	66	CSUSB Main
1002	Mark	Preciado	68	CSUSB Main

Extra Credit Query 1

Description

A student requests to move into an apartment with his friend Melissa Vaughn. Display the Apartment Number and Available Rooms for Melissa Vaughn's Apartment.

SQL

```
/*
A student requests to move into an apartment with his friend Melissa Vaughn. Display the
Apartment Number and Available Rooms for Melissa Vaughn's Apartment.
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: Group
*/

select sa.apartment_number "APT #", sa.available_rooms
from student_apartment sa, room rm, rents rts, student s
where sa.residence_number = rm.residence_number
and rm.room_number = rts.room_number
and rts.student_number = s.student_number
and s.student_number = (select student_number from student where last_name = 'Vaughn');
```

Result

```
APT #  AVAILABLE_ROOMS
------ ---------------
7001                 3
```

Extra Credit Query 2

Description

List all Student Apartments, their rental price and available rooms, ordered by price per room.

SQL

```
/*
List all Student Apartments, their rental price and available rooms, ordered by price per room
Project Option: Coyote Residence Office
TEAM: King James & The Knights of the Data Table
DONE BY: Group
*/

select apartment_number "Apt #", rent "Rent", available_rooms "Rooms Available"
from room r, student_apartment sa join rented_residence rr
        on sa.residence_number = rr.residence_number
where rr.residence_number = r.residence_number
and sa.available_rooms > 0
group by apartment_number, rent, available_rooms
order by rent asc;
```

Result

APT #	RENT	AVAILABLE_ROOMS
7005	350	2
7003	400	1
7006	450	1
7002	500	4
7001	600	3

Extra Credit Query 3

Description

Retrieve the Student Name, Student ID, their next of kin name, phone number and relationship for any student under 21 years of age living in an Student Apartment where an Inspection Report for said apartment had a comment mentioning alcohol.

SQL

```
SQL> get q3.sql
  1  select s.first_name || ' ' || s.last_name "Student", s.student_number "SID", nr.nok_name "Next of Ki
n",n.nok_phone_number "Phone #", nr.nok_relationship "Relationship"
  2  from jsmall.student s, jsmall.nok_relationship nr, jsmall.next_of_kin n, jsmall.invoice i, jsmall.ro
om r, jsmall.rents rts
  3  where to_number(to_char(sysdate, 'YYYY')) - to_number(to_char(19) || to_char(s.date_of_birth, 'YY'))
 < 21
  4  and s.student_number = nr.student_number
  5  and nr.nok_name = n.nok_name
  6  and s.residence_type = 'Student Apartment'
  7  and r.residence_number = (select residence_number from jsmall.student_apartment sa, jsmall.room ro,
jsmall.inspection_report ir where ir.comments like '%alcohol%'
  8     and sa.apartment_number = ir.apartment_number)
  9  and r.room_number = rts.room_number
 10* and rts.student_number = s.student_number
```

Result

```
no rows selected
```

Comments on Project - Normalization Phase

A. Difficulties you faced in doing this implementation phase and how they were resolved?

Some of the difficulties we faced involved just getting all of the work done in a timely manner and creating this document. With so many tables, it took a lot of images used to display things correctly which took time to create and insert. The solution was for James to make one big spool of all of the necessary print outs, and each of us made screenshots from that spool document. Another difficulty we faced was getting the comment tables and table descriptions to display properly, so the information would actually be readable. This was finally accomplished using SQL's built-in commands for display settings, after we searched online and in the provided manuals on how to accomplish this. Another struggle we went through was learning the nuances of how to

write the .sql files needed for this portion of the project. By trial and error, we learned that SQL needed double quotes where we were using single quotes, and when to use commas and semicolons.

B. Likes and dislikes about this part of the project?

Carbajal, Enrique: While working on this phase of the project, I enjoyed creating the tables and creating the comments. I liked creating the tables because we got to implement what we learned in class and apply it to our project. Finally seeing the tables is well worth all the work that was previously done. In addition, I thoroughly enjoyed creating the comments for the tables and columns because I got a little bit of leeway when it came to creating the comments that didn't already exist. Doing the comments that already existed in the data dictionary was a piece of cake. Moreover, for the comments that didn't already exist, I liked writing comments that would help out future users of our database. Last, what I didn't like about this part of the project was the amount of typing that was needed. The information needed for all of my inserts was lines and lines long. I frequently had to take breaks because my fingers and forearms became fatigued.

Small, James: I liked on this part of the project that we got to actually use oracle finally. Creating the tables and putting actual data in made everything we've done up to this point worth it finally. I disliked how much work it took. Part of that is because we have so many tables compared to a lot of groups that it turned out to be a lot of work.

Takahashi, Mark: I did not like creating comments for everything in our database. It was very tedious, however i see that it is good to make notes for everything even if it was obvious. I liked how we got to run queries and use oracle and basically tie in everything we did for each part and complete the project.

Urbach, Daniel: The part I liked most about this part of the project was getting to see all of our hard work over the quarter substantiate into a tangible database. Also queries and the logic involved in correctly creating them is what interests me the most, so this section appealed to me. The part I disliked about this section was creating the comments for each of the tables assigned to me, because it was considerably tedious.

C. What was the most challenging aspect of this part of the design?

Carbajal, Enrique: This part of the project was most tedious. The amount of information that we needed to input into the database became apparent when I began

doing all my insert commands. Because of all the typing I had to do, that was the most difficult part for me.

Small, James: The most challenging aspect was creating the data. With so many foreign key constraints going on, it was difficult to make sure all the data was in alignment so that when we actually did the insert, it all worked. Luckily for us, it worked first try.

Takahashi, Mark: This part of the project was a lot of work and finding time to complete everything in a reasonable manor was difficult. I found the insert commands a little tricky but not very difficult. Typing all the comments for each table and column was straightforward but very time consuming.

Urbach, Daniel: The most challenging part of this section was simply finding time to work as a group and get the project finished. Because it was the end of the quarter, and there was a lot of work for all of us to do and a lot of studying which needed to be done for all of our courses, just being able to convene and collaborate was quite a task.

D. Suggestions on how to improve this part of project?

Carbajal, Enrique: My suggestions on improving the project would be having less amount of typing needed to do. If it were somehow possible to take all the data that we had and insert it into the database, that would help this part of the project tremendously. I made many mistakes with my typing. I also thought that all of my typing was correct. I use a text editor for Ubuntu called Codefish. It color coordinates the lines. Since all of my typing was coming out the correct colors, I thought I was doing it correctly. Only until we tried to run the commands did we find out that most of it was incorrect.

Small, James: Probably the only way I could see to improve this part of the project would be to have less tables. Since it was our design, we can't complain that there were so many tables because we designed it that way, but it turned out to be a lot of work with 17 tables in all. As far as the work itself, the only thing I would change is the comments. We didn't need to demonstrate inserting of comments on 17 tables over 80+ attributes. We could have demonstrated our knowledge on only a couple to prove a point.

Takahashi, Mark: The main suggestion i would have is to give more time to work on this portion of the project. Since there were so many tables we had a lot to type and to ensure that all of our formats were the same across the board.

Urbach, Daniel: The only thing I could suggest for this section would be to break it up into two separate parts: one for table creation, comments and insertion, and a whole separate section for queries, which I feel were a very small part in the project overall.

E. Did you make changes to your communication methods? If yes, why?

Carbajal, Enrique: For the way that our group communicated, we didn't make any changes. No changes were necessary because the channels of communication worked for the first and second part of the project. I didn't' see a need to change this.

Small, James: No changes to the communication. Most of the initial planning was done together, breaking apart who did what. The rest of the communication was over email.

Takahashi, Mark: We did not make any changes to communication. Throughout the entire project we stayed consistent with our communication methods. We split up portions of the project, used email and google drive to collaborate our work.

Urbach, Daniel: No changes to speak of here, the work was apportioned appropriately, and the work was done in the same manner, communicating over email and google chat.