



Qureshi 041063530 CST2355 - Assignment 1 (Auto Recovered)

Database System (Algonquin College)



Scan to open on Studocu

CST2355 - Database Systems

Assignment 1: Setting up Databases (15%)

This assignment relates to the following Course Learning Requirements:

CLR 1 - Plan, Prepare, Install, Configure, and Use a market-leading Database Management System, Data Modeling Engineering Tools, and Open Source Software.

CLR 2 - Develop Advanced Database Design and Normalization

CLR 3 - Develop advanced subjects and techniques of using the SQL database language

Background

You are a new employee at a startup company Van Whinkle. The company just purchased several mom and pop hotels that they want to incorporate under their Van Whinkle brand name. Because the company is new and the hotels were previously owned by small business owners, they do not have any electronic databases. Most have their data on paper. Only a few were a little bit more organized and had spreadsheets and QuickBooks.

Instructions

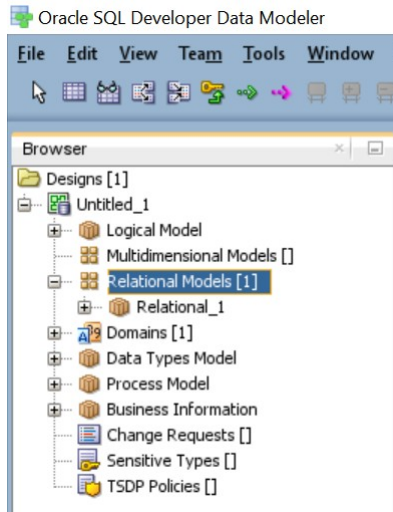
Execute the following steps. Please submit your work only where the instructions specify to do so. If you do not follow instructions, your submission will be rejected.. For this assignment, we will be using the Oracle DBMS with Data Modeler, SQL Developer, and SQLPlus.

	Activity
1	Install Oracle Unlike other tools such as MySQL and SQL Server, the Oracle server and its needed tools are downloaded separately. The following steps downloads the Oracle database. i. If Oracle is already installed then go to step (iv) ii. Locate the most recent version of Oracle from the Oracle website iii. Download and install the Oracle database iv. Confirm all Oracle services are started

	There is no need to create an Oracle database. One is created by default.
2	<h2>Install Oracle Data Modeler</h2> <p>Oracle SQL Developer Data Modeler or Data Modeler is a visual modeling tool used to create your tables and relationships. It's a good practice to use visual modeling. Some of the benefits of visual modeling include --</p> <ul style="list-style-type: none"> - Providing a quick overview of database tables - Generating and regenerating code for multiple databases - Easier to identify changes and differences using diagrams <p>Oracle Data Modeler will be used to create your Database Design Diagrams.</p> <ol style="list-style-type: none"> Download the most recent version Data Modeler Unzip the downloaded file Add a shortcut to the extracted file datamodeler.exe Double-click on the shortcut to start Data Modeler
3	<h2>Install Oracle SQL Developer</h2> <p>Oracle SQL Developer or SQL Developer helps you write and run SQL code. During this assignment Data Modeler will be used to create SQL and SQL Developer will run that SQL.</p> <ol style="list-style-type: none"> Download the most recent version of SQL Developer Unzip the downloaded file Add a shortcut to the extracted file sqldeveloper.exe Double-click on the shortcut to start SQL Developer <p>Note: <i>SQL Developer has a feature that allows you to concurrently use Data Modeler as a part of SQL Developer. Because of difficulties students had using this feature, it is strongly recommended that you use SQL Developer and Data Modeler as separate programs.</i></p>
4	<h2>Add FEE Table to Database Design</h2> <p>You will create a one table database design. This diagram is called a database design and NOT an E-R diagram. Because of the confusion it creates the term E-R diagram will not be used during this course.</p> <ol style="list-style-type: none"> Open Data Modeler Hover your mouse-pointer over each icon near the top of the window to see what it is



iii. Locate the object browser. The visual hierarchy located on the left of the window




iv. Right-click Relational Models in the object browser and select New Relational Model in the menu

v. Right-click the Relational Model you just created and select Show in the menu

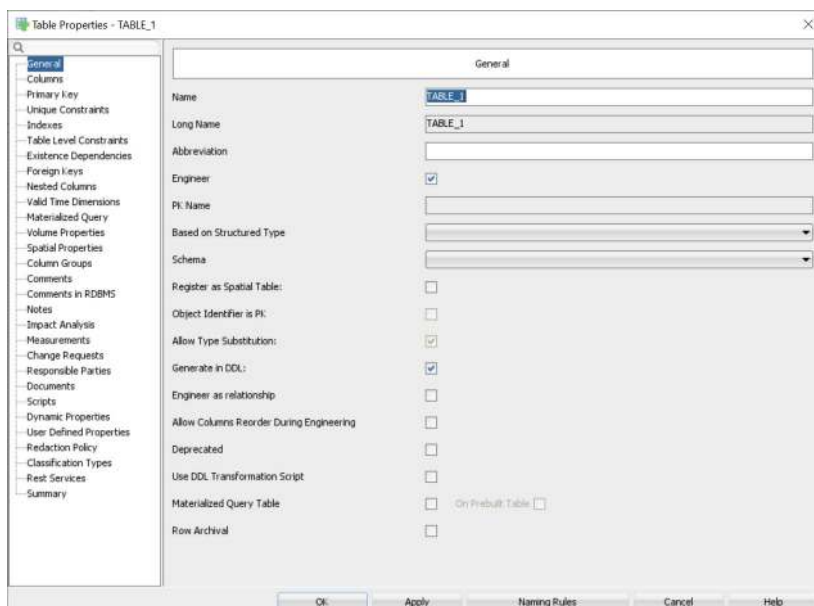
vi. Locate the tabs for each of your worksheets. A Worksheet is the area you work in under a Tab. The tabs look similar to those below.




vii. Locate the Relational Model you created and showed

viii. Click on the Table Icon  then click on any area within your relational model worksheet


ix. You should now see the Table Properties Window. You see this window each time you create a table or double-click on a table in the diagram.





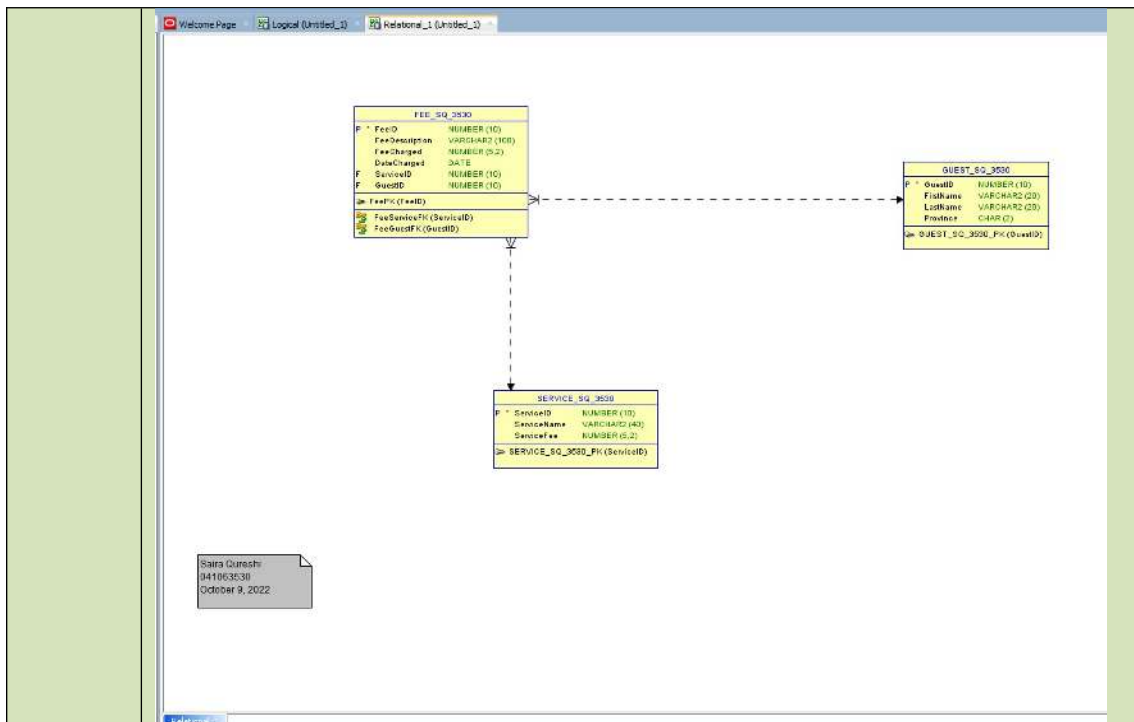
	<p>x. Click on each property group in the browser to the left of the window to view each of the associated properties and what they entail</p> <p>xi. Click <u>General</u> in the property group browser.</p> <p>xii. Name your table <u>FEE_NN_SSSS</u> where NN_SSSS is the initials of your first/last name and the last four digits of your student number. You would enter in your table name to the right of the caption Name.</p> <p style="color: red;">Student Bob Clark with student number 12345678 would name the table <u>FEE BC 5678</u></p> <p>xiii. Click on the <u>Apply</u> button at the bottom of the window</p>				
<h1>5</h1>	<h2>Naming Standards</h2> <p>The common naming standard is having UPPER_CASE table names and ProperCase field names.</p> <table border="1"> <tr> <td>UPPER_CASE</td><td>EMPLOYEE, PRODUCT, CORPORATE_OFFICE</td></tr> <tr> <td>ProperCase</td><td>EmployeeID, LastName, SKU_ID</td></tr> </table> <p>are naming when they are absolutely needed to separate words for readability. For example, SKU_ID is easier to read than SKUID.</p> <p>Note:</p> <p><i>This naming standard applies to all your diagrams and when you write your SQL. Here's an example of the naming for a SELECT statement.</i></p> <p style="color: red;">SELECT FirstName FROM EMPLOYEE;</p> <p><i>When creating your tables in Oracle, however, do not rename the default UPPER_CASE fields to ProperCase. Doing so will require you to add double-quotes to your column names for all your SQL.</i></p> <p style="color: red;">SELECT "FirstName" FROM EMPLOYEE;</p> <p><i>Notice the double quotes. This SQL statement will not run on other databases as is because of these double-quotes.</i></p>	UPPER_CASE	EMPLOYEE, PRODUCT, CORPORATE_OFFICE	ProperCase	EmployeeID, LastName, SKU_ID
UPPER_CASE	EMPLOYEE, PRODUCT, CORPORATE_OFFICE				
ProperCase	EmployeeID, LastName, SKU_ID				
<h1>6</h1>	<h2>Add ID Column</h2> <p>It is common practice today to give each table an ID column. This ID column gives a unique value for each row in the table. The column for the ID is named after the table. For example, in the table EMPLOYEE you would have EmployeeID. And for FEE you would have a FeeID. Having this ID column allows queries and applications to know <u>EXACTLY</u></p>				

	<p>which row of data was either retrieved or modified. To simplifies query and coding logic.</p> <ol style="list-style-type: none"> Click on <u>Columns</u> in the property group browser Click on the <u>Green Plus Icon</u>  to add a column To the field right of <u>Name</u> enter in the value <u>FeeID</u> Click <u>Logical Toggle</u> found on the right of <u>Data Type</u> Select <u>NUMERIC</u> in the Dropdown for <u>Source Type</u> Type <u>10</u> in the field to the right of <u>Precision</u> (total of 10 digits for the number) Type <u>0</u> in the field to the right of <u>Scale</u> (0 digits after the decimal) Click the <u>PK Checkbox</u> to constrain the column values as a primary key Click the <u>Apply Button</u> located at the bottom of the window to save you changes <p>The maximum value for FeeID is 9,999,999,999.</p> <p>Because the ID column is unique for each row, it makes sense to make the ID column the primary key for the table. It's a good practice to specify the size of the primary key to stay aware of the column's limitations and incompatibilities. For instance, for Oracle using the INTEGER data type limits your primary key values to be between -2,147,483,648 and 2,147,483,647. Even small companies can easily hit this limit. Also INTEGER types for one technology may not match well with INTEGER types for another technology. Issues like these are often never discovered until after hours of troubleshooting.</p>
7	<h2>Rename Primary Key Constraint</h2> <p>When you set a column to be a primary key, you add a primary key constraint. The column and the constraint are two different objects in a database.</p> <ol style="list-style-type: none"> Click on <u>Primary Key</u> in the property group browser. Rename <u>FEE_PK</u> to <u>FeePK</u> Click the <u>Apply Button</u>
8	<h2>Add Description Column</h2> <p>It's common for each fee to have a short descriptive name. This descriptive name is often included as a line item in an invoice or receipt. Doing this allows you to keep history of what the fee was called at the time of it being charged. This allows you to rerun past invoice/receipts and reproduce the exact same results. When fee descriptions change over time, the history of those descriptions never change.</p>

	<ul style="list-style-type: none"> i. Click on <u>Columns</u> in the property group browser ii. Click on the <u>Green Plus Icon</u> to add a column and name it <u>FeeDescription</u> iii. Click <u>Logical Toggle</u> for <u>Data Type</u>. iv. Select <u>VARCHAR</u> for <u>Source Type</u> v. Type <u>100</u> for <u>Size</u> (this allow the column to store 20 characters of text) vi. Click the <u>Apply Button</u>
<p>9</p>	<p>Add FeeCharged Column</p> <p>When charging a fee you need to record the fee charged. This is to ensure that when fee changes in the future, that history of what customers were actually charged won't change. This allows you to rerun invoices/receipts at anytime and reproduce the same results.</p> <ul style="list-style-type: none"> i. Click on the <u>Green Plus Icon</u> to add a column and name it <u>FeeCharged</u> ii. Click <u>Logical Toggle</u> for <u>Data Type</u> iii. Select <u>NUMERIC</u> for <u>Source Type</u> iv. Type <u>5</u> for <u>Precision</u> (this allows the column to store 5 digit numbers) v. Type <u>2</u> in the field to the right of <u>Scale</u> (this allows the column to store 2 digits after the decimal and 3 digits before the decimal) vi. Click the <u>Apply Button</u>
<p>10</p>	<p>Add DateCharged Column</p> <p>When charged a fee, you ALWAYS need to include what date the fee was charged. This is so you can confirm fees were charged and paid. It also helps settle client disputes where the client asserts they already paid.</p> <ul style="list-style-type: none"> i. Click on the <u>Green Plus Icon</u> to add a column and name it <u>DateCharged</u> ii. Click <u>Logical Toggle</u> for <u>Data Type</u> iii. Select <u>Date</u> for <u>Source Type</u> iv. Click the <u>Apply Button</u>

11	<h2>Close the Table Properties Window</h2> <p>i. Click the <u>OK Button</u> at the bottom of the window</p>
12	<h2>Add SERVICE Table</h2> <p>The company wants to track what services were charged for which fees. To do this, a service table is created which will be a parent of the fee table.</p> <p>i. Add the SERVICE table. Give it the name SERVICE_NN_SSSS where NN_SSSS is your first/last name initials and the last four digits of your student number.</p> <p>ii. Add the columns ServiceID with NUMERIC(10), ServiceName with VARCHAR(40), and ServiceFee with NUMERIC(5,2)</p> <p>iii. Make ServiceID the primary key</p> <p>iv. Click  icon</p> <p>v. Click the SERVICE (parent) table then click the FEE (child) table to draw a relationship line between the two tables</p> <p>vi. In the <u>Foreign Key Properties</u> window un-select <u>Mandatory</u> so you can leave the foreign key column blank. Rename the constraint FeeServiceFK.</p> <p>vii. Confirm the foreign key column SERVICE_ServiceID was created on the FEE table</p> <p>viii. Rename the field from SERVICE_ServiceID to ServiceID.</p>
13	<h2>Add GUEST Table</h2> <p>The company wants to track what guests were charged which fees. So you create a guest table that will be another parent of the fee table. You do not add guest information to your FEE table. The FEE table is only for fees and foreign keys to other tables. There is no other information.</p> <p>i. Add the GUEST table. Give it the name GUEST_NN_SSSS where NN_SSSS is your first/last name initials and the last four digits of your student number.</p> <p>ii. Add the columns GuestID with NUMERIC(10), FirstName with VARCHAR(20), LastName with VARCHAR(20), City with VARCHAR(20), and Province CHAR(2)</p> <p>iii. Make GuestID the primary key</p>

	<p>iv. Draw a relationship between GUEST and FEE</p> <p>v. In the <u>Foreign Key Properties</u> window un-select <u>Mandatory</u> so you can leave the foreign key column blank. Rename the constraint FeeGuestFK.</p> <p>vi. Confirm the foreign key column GUEST_GuestID was created on the FEE table</p> <p>vii. Rename the field GUEST_GuestID to GuestID</p>
14	<p>View the Database Design</p> <p>i. Confirm that you have correct columns and constraint for each of your tables</p> <p>ii. Click on the <u>FEE</u> table and resize the table to remove white space and avoid clipping of text</p> <p>iii. Examine the table and notation for the FEE table</p> <p>iv. The  means FeeID is the column used in the Primary Key Constraint</p> <p>v. The  means FeePK is the <u>Primary Key Constraint</u> (constraints are found in the lower box for the table)</p> <p>vi. Add your name, student number and current date to the bottom left corner of your diagram</p>
15	<p>Submit Database Design</p> <p>Add below a screen shot of your Database Design diagram. To receive marks your diagram must includes your tables, name, and student number with the current date Exclude menus, toolbars, and object browsers</p>



16

Forward Engineer Diagram

These steps generates SQL from your diagram. Make sure you save your diagram first. Some tools require your diagram to be saved before generating code.

- i. Select File Menu located at the top of the window
- ii. Select Save/Save As to save your whole design (saves as a .DMD file along with a folder of the same name)
- iii. Locate your saved file and confirm it was saved.
- iv. Select File Menu then Close/All
- v. Select File Menu then Open to select the DMD file you previously saved
- vi. Click on the tab for the Relational Model.
- vii. If you do not see the tab, right-click on the Relational Model in the object browser and select Show
- viii. Click the File Menu then Export then select DDL File
- ix. Click the Generate Button located near the upper right corner
- x. In the DDL Generation Options Window, click on the 'Create' Section tab
- xi. Uncheck all options.
- xii. Check Not Assigned to Schemas (so you can run this script using any schema)

	<p>xiii. In the <u>DDL Generation Options</u> Window, click '<u>Drop</u>' Section tab</p> <p>xiv. Check all options (to drop/recreate your objects when rerunning your script)</p> <p>xv. Click the <u>OK Button</u> located at the bottom of the window</p>
17	<h2>Modify DDL File</h2> <p>i. Inspect the contents of the <u>DDL File Editor</u> window</p> <p>ii. Remove all lines that prefix with hyphens (these are comments and are not needed)</p> <p>iii. Add a comment on the first line that includes your name, student number, and current date. For Bob Clark the comment could look like...</p> <p style="text-align: center;">-- Bob Smith, 12345678, 2022/07/01</p> <p>iv. You should have the SQL statements starting off with: DROP, CREATE TABLE, ADD CONSTRAINT</p> <p>v. Rewrite this script so table names are UPPER_CASE and columns are ProperCase (this allows you to reuse the script to run on different databases)</p> <p>vi. Remove <u>CASCADE CONSTRAINTS</u> from the <u>DROP</u> statement by making the statement look like the following</p> <p style="text-align: center;">DROP TABLE TABLE_NAME;</p> <p style="text-align: center;">As a practice you drop one constraint at a time so you can easily backtrack and undo mistakes</p> <p>vii. Merge the ALTER TABLE statement with the CREATE TABLE following the below patterns (this improves the readability and understandability of the table)</p> <pre>CREATE TABLE TABLE_NAME (TableNameID, ... LastField, CONSTRAINT TableNamePK PRIMARY KEY (TableNameID));</pre> <p style="text-align: center;">OR</p> <pre>CREATE TABLE TABLE_NAME (TableNameID, ... LastField, CONSTRAINT TableNamePK PRIMARY KEY (TableNameID), CONSTRAINT ConstraintName FOREIGN KEY (ParentID) REFERENCES ChildTableName (ChildTableID));</pre>

	<p>viii. To avoid compile errors, verify you have a comma after the last field, a common after every CONSTRAINT clause, and no comma/semicolon after the last (TableNameID) / (ChildTableID) in the CREATE TABLE statement. There must be a closing bracket at the end of your CREATE TABLE statement.</p> <p>ix. Reorder the DROP statements so the child table (FEE) is dropped before the parent tables are (SERVICE and GUEST). Parents cannot be successfully dropped before their children.</p> <p>x. Reorder the CREATE statements so the child table (FEE) is created after the parent tables are (SERVICE and GUEST). Children cannot be successfully created before their parents.</p> <p>xi. Verify you have a semicolon after the closing bracket for the CREATE TABLE statement</p> <p>xii. Click the <u>Save Button</u> at the bottom of the window to save your DML file</p>
18	<h3>Add ASSIGN_1 User</h3> <p>Skip this activity if you already added your ASSIGN_1 user using SQLPlus.</p> <ol style="list-style-type: none"> Go to the search bar at the bottom left of your desktop Type <code>cmd</code> and hit <u>ENTER</u> to open the DOS window Type <code>SQLPLUS \ AS SYSDBA</code> then hit <u>ENTER</u> (this logs you in as a SYSDBA for super user privileges) If prompted for a password, hit <u>ENTER</u> Type <code>ALTER SESSION SET "_ORACLE_SCRIPT"=true;</code> then hit <u>ENTER</u> to set your session so you can add users Type <code>CREATE USER ASSIGN_1 IDENTIFIED BY 1234;</code> then hit <u>ENTER</u> to add user assign_1 with the password 1234 Type <code>GRANT ALL PRIVILEGES TO ASSIGN_1;</code> then hit <u>ENTER</u> to give the user all permissions Close the DOS window.
19	<h3>Create tables FEE, GUEST, and SERVICE</h3> <p>This activity will have you use SQLPLUS to create your tables</p> <ol style="list-style-type: none"> Go to the search bar at the bottom left of your desktop

	<p>ii. Type <u>cmd</u> and hit <u>ENTER</u> to open the DOS window</p> <p>iii. Type <u>SQLPLUS</u> then hit <u>ENTER</u></p> <p>iv. Type your ASSIGN_1 user name then hit <u>ENTER</u></p> <p>v. Type your password for user ASSIGN_1 then hit <u>ENTER</u></p> <p>vi. Copy and paste the contents of the DDL file you modified at the SQLPLUS prompt then hit ENTER</p> <p>vii. Create a screen shot of your DOS window that includes your name, student number, date, SQL statements, and the script results</p>
--	--

20

Submit Forward Engineering Script

Add below a screen shot of **ONLY** your DOS window To receive marks your screenshot must includes all SQL statements with its results along with your name, student number, and current date

```

C:\Users\Saira>SQLPLUS
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Saira>SQLPLUS

SQL*Plus: Release 19.0.0.0.0 - Production on Tue Oct 11 00:29:16 2022
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Enter user-name: ASSIGN_1
Enter password:
Last Successful login time: Tue Oct 04 2022 23:25:46 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> -- Saira Qureshi, 041063530, 2202/10/09
SQL>
SQL> DROP TABLE FEE_SQ_3530;
DROP TABLE FEE_SQ_3530
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> DROP TABLE GUEST_SQ_3530;
DROP TABLE GUEST_SQ_3530
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> DROP TABLE SERVICE_SQ_3530;
DROP TABLE SERVICE_SQ_3530
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> CREATE TABLE GUEST_SQ_3530 (
  2   GuestId  NUMBER(10) NOT NULL,
  3   FirstName VARCHAR2(20),
  4   LastName  VARCHAR2(20),
  5   Province  CHAR(2),
  6
SQL>      CONSTRAINT GUEST_SQ_3530_PK PRIMARY KEY ( GuestId )
SP2-0734: unknown command beginning "CONSTRAINT..." - rest of line ignored.
SQL> );
SP2-0842: unknown command ")" - rest of line ignored.
SQL>
SQL> CREATE TABLE SERVICE_SQ_3530 (
  2   serviceid  NUMBER(10) NOT NULL,
  3   servicename VARCHAR2(40),
  4   servicefee  NUMBER(5, 2),
  5
SQL>      CONSTRAINT SERVICE_SQ_3530_PK PRIMARY KEY ( ServiceId )
SP2-0734: unknown command beginning "CONSTRAINT..." - rest of line ignored.

```

```

SQL> Select Command Prompt - SQLPLUS
SP2-0734: unknown command beginning "CONSTRAINT..." - rest of line ignored.
SQL> );
SP2-0042: unknown command ")" - rest of line ignored.
SQL>
SQL> CREATE TABLE SERVICE_SQ_3530 (
  2   serviceid NUMBER(10) NOT NULL,
  3   servicename VARCHAR2(40),
  4   servicefee NUMBER(5, 2),
  5
SQL>   CONSTRAINT SERVICE_SQ_3530_PK PRIMARY KEY ( ServiceId )
SP2-0734: unknown command beginning "CONSTRAINT..." - rest of line ignored.
SQL> );
SP2-0042: unknown command ")" - rest of line ignored.
SQL>
SQL> CREATE TABLE FEE_SQ_3530 (
  2   FeeId NUMBER(10) NOT NULL,
  3   FeeDescription VARCHAR2(100),
  4   FeeCharged NUMBER(5, 2),
  5   DateCharged DATE,
  6   ServiceId NUMBER(10),
  7   GuestId NUMBER(10),
  8
SQL>   CONSTRAINT FeePK PRIMARY KEY ( FeeId )
SP2-0734: unknown command beginning "CONSTRAINT..." - rest of line ignored.
SQL> );
SP2-0042: unknown command ")" - rest of line ignored.
SQL>
SQL> ALTER TABLE " FEE_SQ_3530"
  2   ADD CONSTRAINT " FeeGuestFK" FOREIGN KEY ( guestid )
  3   REFERENCES guest_sq_3530 ( guestid );
ALTER TABLE " FEE_SQ_3530"
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> ALTER TABLE " FEE_SQ_3530"
  2   ADD CONSTRAINT " FeeServiceFK" FOREIGN KEY ( serviceid )
  3   REFERENCES service_sq_3530 ( serviceid );
ALTER TABLE " FEE_SQ_3530"
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL>
SQL>
SQL>

```


21

Connecting to Oracle in SQL Developer

i. Open [SQL Developer](#)

ii. Hover the mouse-pointer over each icon in the upper left corner of the window to see what each icon does



iii. Click on the  icon to add a connection


- iv. Type Assign1 for Name
- v. Type ASSIGN_1 for Username
- vi. Type your password for Password
- vii. Click SID Toggle
- viii. Type ORCL for SID
- ix. Click the Test Button at the bottom of the window to test your connection
- x. Confirm the test is successful by viewing the status at the bottom right corner of the window
- xi. Click the Save Button at the bottom of the window to save your connection
- xii. Click the Connect Button at the bottom of the window to connect to the database
- xiii. When prompted for your password, type in your password
- xiv. Click the Assign1 Worksheet Tab to select it
- xv. Hover your mouse-pointer over each icon under the worksheet tab to see what each one does



22

View Table Structures

By default Oracle creates field names in UPPER_CASE. Do not rename these columns in ProperCase. If you do, it forces you add double-quotes to all column names in your SQL. This hardwires your SQL to Oracle and that SQL not work as is on other databases. Keep all columns created in Oracle in UPPER_CASE to ensure portability.

- i. Click on the FEE table in the object browser
- ii. Click on the FEE Tab
- iii. Click on SQL Sub-Tab located under the FEE Tab
- iv. Examine the script and identify all differences it has from your DDL file (**these differences are Oracle specifics**)
- v. Click on Columns Sub-Tab (**this is your table designer window**)
- vi. Examine each column row
- vii. Click on the  to add a column
- i. Name the new column with your first and last name initials

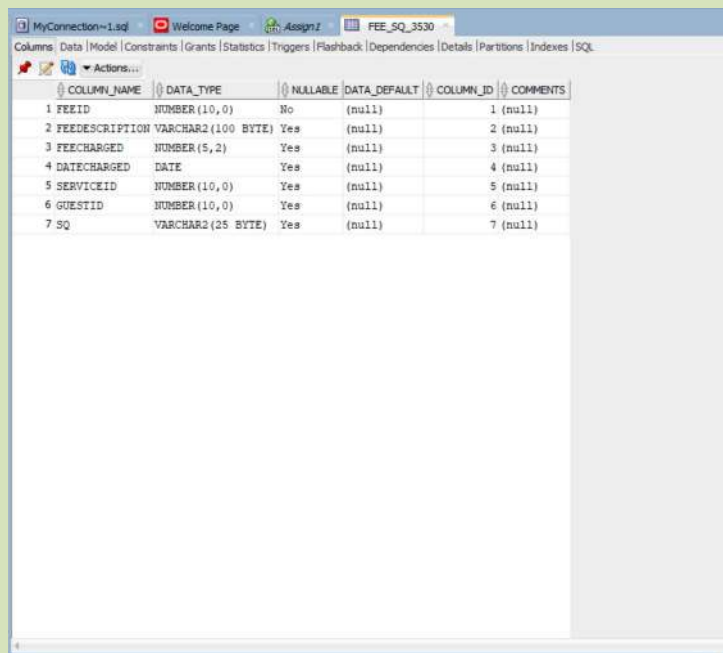
viii. Click the OK Button

ix. Create a screen shot of table designer (excluding menus, toolbars, and object browsers)

23

Submit Table Designer Screen Shot

Add below a screen shot of the FEE table design. To receive marks your screenshot must includes your table name and all of its columns and field attributes.



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	FEEID	NUMBER (10, 0)	No	(null)	1	(null)
2	FEEDescription	VARCHAR2 (100 BYTE)	Yes	(null)	2	(null)
3	FEECHARGED	NUMBER (5, 2)	Yes	(null)	3	(null)
4	DATECHARGED	DATE	Yes	(null)	4	(null)
5	SERVICEID	NUMBER (10, 0)	Yes	(null)	5	(null)
6	GUESTID	NUMBER (10, 0)	Yes	(null)	6	(null)
7	SQ	VARCHAR2 (25 BYTE)	Yes	(null)	7	(null)

24

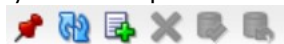
Add GUEST Data


i. Click on the GUEST table in the object browser

ii. Click on the GUEST Tab

iii. Click on Data Sub-Tab

iv. Hover your mouse-pointer over each icon under the Data Sub-Tab to see what it does




v. Click  icon to insert a row

vi. Type the value 1 in GuestID then press TAB

vii. Type your first name in FirstName then press TAB


viii. Type your last name in LastName then press TAB


ix. Type today's date in City

x. Click the  Icon to commit your changes (you cannot undo after committing)

xi. Add and commit the following data to GUEST each with their own GuestID (please use the two letter codes for each province)

- Kristoff Kurn lives in Vancouver, British Columbia
- Billy Elliot lives in Winnipeg, Manitoba
- Tanya Duncan works for the hotel and lives in Woodstock, Ontario
- Fred MacFadden lives in Ottawa, Ontario
- Deena Donor works for the hotel and lives in Toronto, Ontario
- Justin Hackman lives in London, Ontario.

xii. If you make any mistakes, use  to delete select rows

xiii. If you need to redo the whole task over, use  Icon to reverse all uncommitted changes

You keep all guest names and information in the GUEST table. You do not duplicate this information in any other table. The more duplicates of information, the more processing power and network bandwidth you need to use to ensure all duplicates are exactly the same. For each duplicate you face a cost.

Activity	Cost
Managing where duplicates are located	More documentation and more scripts that need to be maintained and troubleshooted
Ensuring all duplicates are exact copies	Maintaining scheduled batches and triggers with complex error handling to ensure data is kept in sync
Avoiding dirty reads	Adding logic and constraints to ensure the duplicated data being read is correct and up-to-date.

25



Add SERVICE Data

i. Click on the SERVICE table in the object browser

ii. Click on the SERVICE Tab

iii. Click on Data Sub-Tab

iv. Click  icon to insert a row

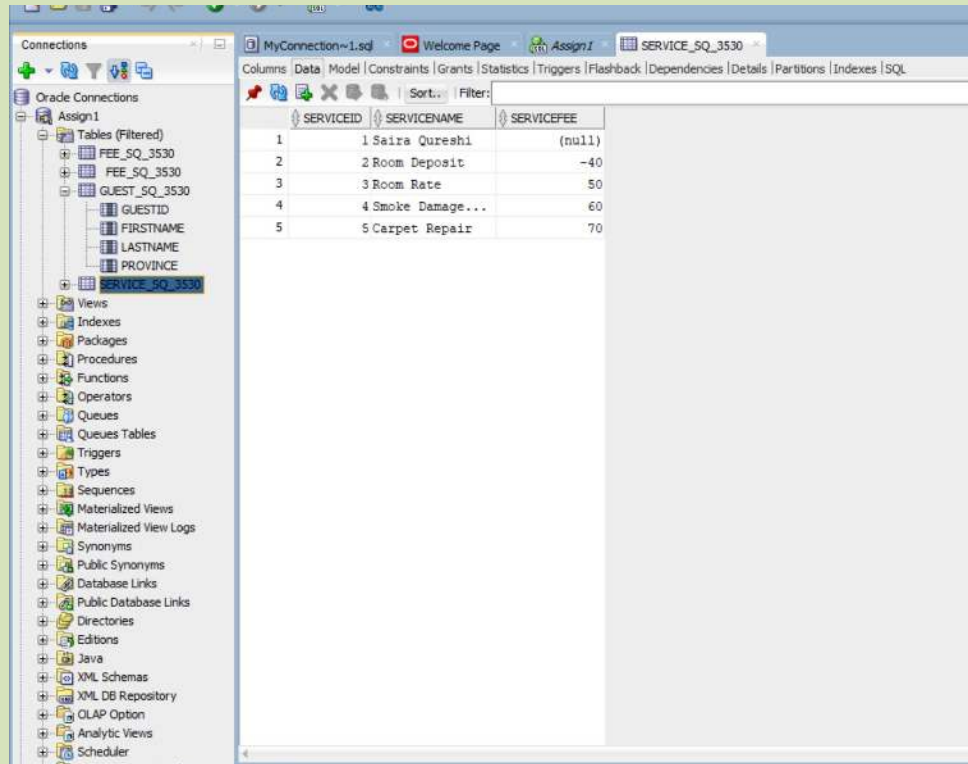
	<p>v. Type the value <u>1</u> in <u>ServiceID</u></p> <p>vi. Type your name in <u>ServiceName</u></p> <p>vii. Commit your changes</p> <p>viii. Add and commit the following data to SERVICE each with their own ServiceID</p> <ul style="list-style-type: none"> - Room Deposit, -\$40 (deposit is a negative amount) - Room Rate, \$50 - Smoke Damage Repair, \$60 - Carpet Repair, \$70
<h1>26</h1>	<h2>Add FEE Data</h2> <p>i. Go to the <u>Data Sub-Tab</u> for <u>FEE</u></p> <p>ii. Type the value <u>1</u> in <u>FeeID</u></p> <p>iii. Type your name in <u>FeeDescription</u> then press <u>TAB</u></p> <p>iv. Type the value <u>0.00</u> in <u>FeeCharged</u></p> <p>v. Double-click on the DateCharged field to see the  icon</p> <p>vi. Click the  icon to set <u>DateCharged</u> to today's date then click <u>OK</u></p> <p>vii. Commit your changes</p> <p>viii. Add and commit the following data to FEE each with their own FeeID. Associate each fee to the correct guest and service using IDs. Do not leave any columns blank.</p> <p><u>Kristoff Kurn</u> was charged for Feb 3rd to Feb 17th. She paid a deposit of \$50 on Feb 3rd. (store one row for the deposit charged). Her room rate per day was \$50 from Feb 3rd to Feb 10th. (store 8 rows of room rates charged) And her room rate was \$45 from Feb 11th to Feb 17th. (store 7 rows of the room rate charged). She was a smoker so she was charged \$25 for smoke damage and \$30 for the carpet burns on Feb 17th.</p> <p><u>Billy Elliot</u> was charged last year on Jan 20th. He is a student at Queens University and paid the student discount rate of \$30 per day. Because he is a loyal guest he didn't need to pay a deposit on Jan 20th. (this means you need to store a record showing the guest was charged a deposit of 0.00)</p> <p><u>Justin Hackman</u> paid last year a discount rate of \$35 per day and a deposit of \$20 because he is a smoker. He was charged the room rate for Feb 2nd, 8th, 17th, and 28th</p> <p><u>Deena Donor</u> was charged last year on the 1st of every month from January through to April. Because of her position, she is not charged a deposit or room rate. (this means you need to record the room rate being 0.00 and the deposit</p>

being 0.00 for each 1st of the month)

27

Submit SERVICE Data Tab Screen Shot

Add below a screen shot of the SERVICE data using the table's sub-tabs. To receive marks your screenshot must includes the table name and all the table's column data.



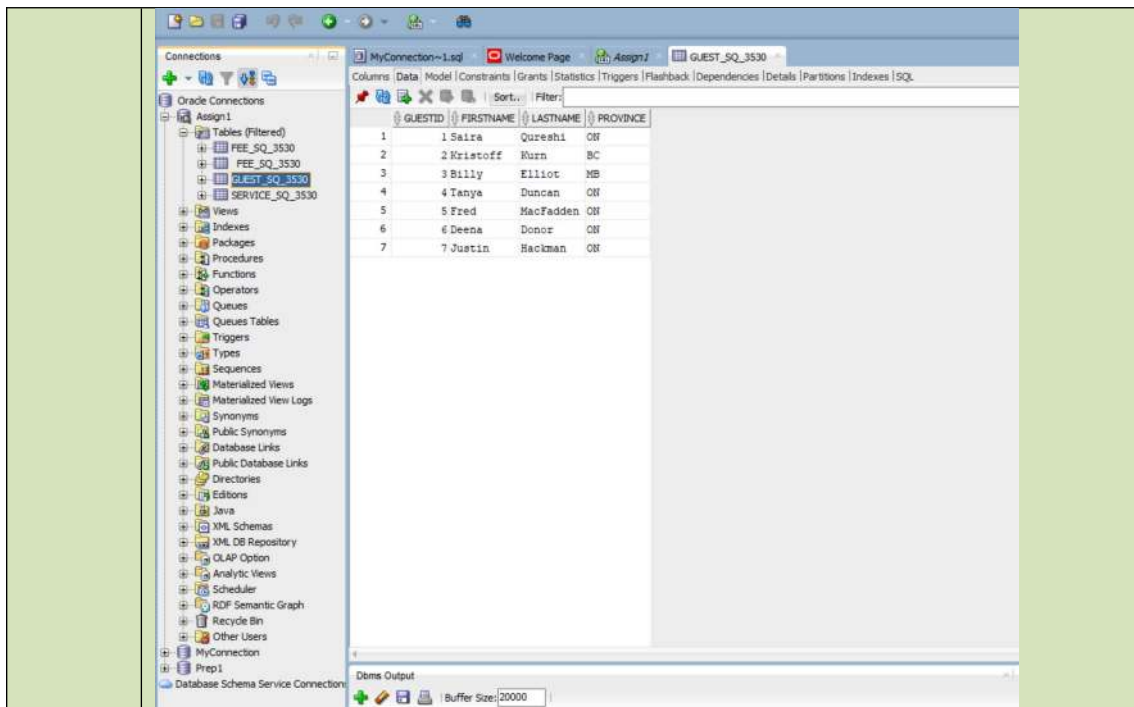
The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane displays a tree view of database objects. The 'Tables (Filtered)' folder is expanded, showing several tables including 'FEE_SQ_3530', 'GUEST_SQ_3530', and 'SERVICE_SQ_3530'. The 'SERVICE_SQ_3530' table is selected. The main window displays the 'Data' tab for this table, showing a list of service records with columns: SERVICEID, SERVICENAME, and SERVICEFEE. The data is as follows:

SERVICEID	SERVICENAME	SERVICEFEE
1	1 Saira Qureshi	(null)
2	2 Room Deposit	-40
3	3 Room Rate	50
4	4 Smoke Damage...	60
5	5 Carpet Repair	70

28

Submit GUEST Data Tab Screen Shot

Add below a screen shot of the GUEST data using the table's sub-tab. To receive marks your screenshot must includes the table name and all the table's column data.




29

Submit FEE Data Tab Screen Shot

Add below multiple screen shots of the FEE data using the table's sub-tab. To receive marks your screenshot must includes the table name and all the table's column data.

FEEID	FEEDESCRIPTION	FEECHARGED	DATECHARGED	SERVICEID	GUESTID	SQ
1	Seira Qureshi	0	22-10-11	(null)	(null)	(null)
2	Deposit	-50	22-02-03	2	2	(null)
3	Room Rate	50	22-02-03	3	2	(null)
4	Room Rate	50	22-02-04	3	2	(null)
5	Room Rate	50	22-02-05	3	2	(null)
6	Room Rate	50	22-02-06	3	2	(null)
7	Room Rate	50	22-02-07	3	2	(null)
8	Room Rate	50	22-02-08	3	2	(null)
9	Room Rate	50	22-02-10	3	2	(null)
10	Room Rate	45	22-02-11	3	2	(null)
11	Room Rate	45	22-02-12	3	2	(null)
12	Room Rate	45	22-02-13	3	2	(null)
13	Room Rate	45	22-02-14	3	2	(null)
14	Room Rate	45	22-02-15	3	2	(null)
15	Room Rate	45	22-02-16	3	2	(null)
16	Room Rate	45	22-02-17	3	2	(null)
17	Deposit	0	21-01-20	2	3	(null)
18	Room Rate	30	21-01-20	3	3	(null)
19	Deposit	-20	22-02-02	2	7	(null)
20	Room Rate	35	22-02-02	3	7	(null)
21	Room Rate	35	22-02-08	3	7	(null)
22	Room Rate	35	22-02-17	3	7	(null)
23	Room Rate	35	22-02-28	3	7	(null)
24	Deposit	0	22-01-01	2	6	(null)
25	Room Rate	0	22-01-01	3	6	(null)
26	Deposit	0	22-02-01	2	6	(null)
27	Room Rate	0	22-02-01	3	6	(null)
28	Deposit	0	22-03-01	2	6	(null)
29	Room Rate	0	22-03-01	3	6	(null)
30	Deposit	0	22-04-01	2	6	(null)
31	Room Rate	0	22-04-01	3	6	(null)

30

	<h2>Generate INSERT Statements</h2> <p>It's a good practice to always regenerate your tables from your database design diagram. The problem with doing this is that after recreating your tables you lose all the data you entered in. To avoid this, you generate a INSERT script to restore all your data in tables after those tables are recreated. You do this using the export feature.</p> <p>The export feature can be done on a single table by right-clicking on that table in the object browser. The export can be done on multiple tables using the <u>Tools</u> option in the menu.</p> <ol style="list-style-type: none"> Right click on the FEE table in the object browser. Select the <u>Column</u> option then the <u>Drop</u> option. Select your <u>name initials</u> column then click <u>Apply</u>. Click on the <u>Tools option</u> in the menu to see the dropdown Click <u>Database Export</u> in the dropdown Select the database connection to use for <u>Connection</u> Un-select <u>Export DDL</u> Select <u>Export Data</u> if it is not already selected Select <u>insert</u> for <u>Format</u> if it is not already selected Select the destination file for your INSERT script using the <u>Browse</u> button Keep clicking the <u>Next</u> button until you have the option to click <u>Finish</u> Click the <u>Finish</u> button You will now see the INSERT statement script in a <u>Worksheet Tab</u> Hit <u>Control-F</u> to view the <u>Find</u> bar Click the  to view the <u>Replace</u> bar Using replace to remove all instances of "ASSIGN_1." (remember to remove the period too) Move all insert statements for your FEE table to the end of your file. Save you worksheet by closing the <u>Worksheet Tab</u>
<h1>31</h1>	<h2>Add STAY Table</h2> <p>A stay for a guest at a hotel is determined not by when they are charged but by when they formally check-in and check-out. Each time a guest checks-in, the database generates a new stay for the guest and associates all fees charged to that stay.</p>

	<p>i. Go to your database design in Data Modeler</p> <p>ii. Add a STAY table. Give it the name STAY_NN_SSSS where NN_SSSS is your first/last name initials with the last four digits of your student number.</p> <p>iii. Add an ID column, CheckinDate as Date and CheckoutDate as Date</p> <p>iv. Add the foreign key relationships where STAY is the parent of FEE and GUEST is the parent of STAY. Make sure all Mandatory options are un-selected and that all foreign keys are created in the correct tables. Rename these constraints FeeStayFK and FeeGuestFK respectfully.</p> <p>v. Remove all table names in the created fields.</p>
32	<p>Add EMPLOYEE Table</p> <p>Employees charge guest fees. Unlike the other tables so far, the employee table uses a recursive relationship. A recursive relationship is where a table references itself. They are used to show hierarchies of projects, teams, or people. For employees, each employee reports to a manager. That manager, too, is an employee. Recursive relationships reduce tables and simplify queries since you don't need separate tables for employees, managers, and their managers.</p> <p>i. Go to your database design in Data Modeler</p> <p>ii. Add the EMPLOYEE table. Give it the name EMPLOYEE_NN_SSSS where NN_SSSS is the first/last name initials with the last four digits of your student number.</p> <p>iii. Add an ID column, FirstName as VARCHAR(20), LastName as VARCHAR(20), Title as VARCHAR(20), City as VARCHAR(20), Province as CHAR(2), Wage as NUMERIC(8,2), and WageType as VARCHAR(20)</p> <p>iv. Add the foreign key relationships where EMPLOYEE is the parent of EMPLOYEE. Make sure the Mandatory option is un-selected and that the foreign key is created in the correct table. Rename the constraint EmployeeFK. Rename the created field to ManagerID.</p> <p>v. Add the foreign key relationships where EMPLOYEE is the parent of FEE. Make sure the Mandatory option is un-selected and that the foreign key is created in the correct table. Rename the constraint FeeEmployeeFK. Rename the created field to EmployeeID.</p>
33	<p>Add VENUE Table</p> <p>A venue is an organization where a guest can go and receive services. It can be a different hotel, casino, spa, or golf course.</p> <p>i. Go to your database design in Data Modeler</p>

	<p>ii. Add the VENUE table. Give it the name VENUE_NN_SSSS where NN_SSSS is your first/last name initials with the last four digits of your student number.</p> <p>iii. Add an ID column and VenueName as VARCHAR(20)</p> <p>iv. Add relationships so VENUE is the parent of all tables except for GUEST and SERVICE. Make sure to un-select Mandatory. Rename the constraints based on the standard used in the previous steps. Rename all created columns to VenueID.</p>
34	<h3>Add Employee is Guest Relationship</h3> <p>Since it's common for people to have the same first and last name, rarely you should use those two columns to determine if someone's the same person. Bob Smith may not be the same Bob Smith. Instead you use an ID relationship.</p> <p>i. Go to your database design in Data Modeler</p> <p>ii. Add the foreign key relationships where GUEST is the parent of EMPLOYEE. Make sure the Mandatory option is un-selected and that the foreign key is created in the correct table. Rename the constraint EmployeeGuestFK. Rename the created field to GuestID.</p>
35	<h3>Forward Engineer and Re-Insert Data</h3> <p>You will now recreate your database from the diagram resulting in you deleting all of the data you already entered in. Make sure you have your INSERT script you created from the previous step available so you can re-run it.</p> <p>i. Forward engineer your database design (remember set it up to drop all tables before recreating them)</p> <p>ii. Go to SQLPLUS</p> <p>iii. Execute a comment including your name, student number, and current date</p> <p>iv. Paste and execute your forward engineering script to recreate your tables. DO NOT edit your DDL file. Run it as is.</p> <p>v. Paste and execute the INSERT script to re-add your GUEST, FEE, and SERVICE data</p> <p>vi. Execute commit; to ensure you can view your inserted data.</p>
36	<h3>Submit Forward Engineering and Insert Script</h3> <p>Add below multiple screen shots of ONLY your DOS window To receive marks your</p>

screenshots must include all SQL statements with its results along with your name, student number, and current date

```
ET Command Prompt - SQLPLUS
C:\Users\Saira\SQLPLUS

SQL*Plus: Release 19.0.0.0.0 - Production on Wed Oct 12 19:12:16 2022
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Enter user-name: ASSIGNS_1
Enter password:
Last Successful login time: Wed Oct 12 2022 18:15:50 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> ::Saira Qureshi 041063530 2022/10/10
SP2-0734: unknown command beginning "::Saina Qu..." - rest of line ignored.
SQL> RER Saira Qureshi 041063530 2022/10/10
SQL>
SQL> DROP TABLE "EMPLOYEE_SQ-3530" CASCADE CONSTRAINTS;
DROP TABLE "EMPLOYEE_SQ-3530" CASCADE CONSTRAINTS
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> DROP TABLE " FEE_SQ_3530" CASCADE CONSTRAINTS;
Table dropped.

SQL>
SQL> DROP TABLE guest_sq_3530 CASCADE CONSTRAINTS;
Table dropped.

SQL>
SQL> DROP TABLE service_sq_3530 CASCADE CONSTRAINTS;
Table dropped.

SQL>
SQL> DROP TABLE stay_sq_3530 CASCADE CONSTRAINTS;
DROP TABLE stay_sq_3530 CASCADE CONSTRAINTS
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> DROP TABLE venue_sq_3530 CASCADE CONSTRAINTS;
DROP TABLE venue_sq_3530 CASCADE CONSTRAINTS
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL>
SQL>
SQL> CREATE TABLE "EMPLOYEE_SQ-3530" (
  2 employeeid NUMBER(20) NOT NULL,
```

```
ET Command Prompt - SQLPLUS

SQL> CREATE TABLE "EMPLOYEE_SQ-3530" (
  2 employeeid NUMBER(20) NOT NULL,
  3 firstname VARCHAR2(20),
  4 lastname VARCHAR2(20),
  5 title VARCHAR2(20),
  6 city VARCHAR2(20),
  7 province CHAR(2),
  8 wage NUMBER(8, 2),
  9 hometype VARCHAR2(20),
  10 venueid NUMBER(20),
  11 guestid NUMBER(10)
  12 );

Table created.

SQL>
SQL> ALTER TABLE "EMPLOYEE_SQ-3530" ADD CONSTRAINT employespk PRIMARY KEY ( employeeid );
Table altered.

SQL>
SQL> CREATE TABLE " FEE_SQ_3530" (
  2 feeid NUMBER(10) NOT NULL,
  3 feedescription VARCHAR2(100),
  4 feecharged NUMBER(5, 2),
  5 datecharged DATE,
  6 serviceid NUMBER(10),
  7 guestid NUMBER(10),
  8 stayid NUMBER(10),
  9 employeeid NUMBER(20),
  10 venueid NUMBER(20)
  11 );

Table created.

SQL>
SQL> ALTER TABLE " FEE_SQ_3530" ADD CONSTRAINT feeapk PRIMARY KEY ( feeid );
Table altered.

SQL>
SQL> CREATE TABLE guest_sq_3530 (
  2 guestid NUMBER(10) NOT NULL,
  3 firstname VARCHAR2(20),
  4 lastname VARCHAR2(20),
  5 province CHAR(2)
  6 );

Table created.

SQL>
SQL> ALTER TABLE guest_sq_3530 ADD CONSTRAINT guestpk PRIMARY KEY ( guestid );
Table altered.

SQL>
SQL> CREATE TABLE service_sq_3530 (
  2 serviceid NUMBER(10) NOT NULL,
  3 servicename VARCHAR2(40),
  4 servicefee NUMBER(5, 2)
  5 );
```



```

SQL> ALTER TABLE service_sq_3530 ADD CONSTRAINT servicepk PRIMARY KEY ( serviceid );
Table altered.

SQL>
SQL> CREATE TABLE stay_sq_3530 (
  2   stayid      NUMBER(10) NOT NULL,
  3   checkindate DATE,
  4   checkoutdate DATE,
  5   guestid     NUMBER(10),
  6   venueid     NUMBER(20)
  7 );
Table created.

SQL>
SQL> ALTER TABLE stay_sq_3530 ADD CONSTRAINT staypk PRIMARY KEY ( stayid );
Table altered.

SQL>
SQL> CREATE TABLE venue_sq_3530 (
  2   venueid     NUMBER(20) NOT NULL,
  3   venuename   VARCHAR2(20)
  4 );
Table created.

SQL>
SQL> ALTER TABLE venue_sq_3530 ADD CONSTRAINT venuepk PRIMARY KEY ( venueid );
Table altered.

SQL>
SQL> ALTER TABLE "EMPLOYEE_SQ_3530"
  2   ADD CONSTRAINT employeguestfk FOREIGN KEY ( guestid )
  3   REFERENCES guest_sq_3530 ( guestid );
Table altered.

SQL>
SQL> ALTER TABLE "EMPLOYEE_SQ_3530"
  2   ADD CONSTRAINT employevenuefk FOREIGN KEY ( venueid )
  3   REFERENCES venue_sq_3530 ( venueid );
Table altered.

SQL>
SQL> ALTER TABLE " FEE_SQ_3530"
  2   ADD CONSTRAINT " FeeEmployeefk" FOREIGN KEY ( employeid )
  3   REFERENCES "EMPLOYEE_SQ_3530" ( employeid );
Table altered.

SQL>
SQL> ALTER TABLE " FEE_SQ_3530"
  2   ADD CONSTRAINT " FeeGuestfk" FOREIGN KEY ( guestid )

```

```

SQL> ALTER TABLE "EMPLOYEE_SQ_3530"
  2   ADD CONSTRAINT employeguestfk FOREIGN KEY ( guestid )
  3   REFERENCES guest_sq_3530 ( guestid );
Table altered.

SQL>
SQL> ALTER TABLE "EMPLOYEE_SQ_3530"
  2   ADD CONSTRAINT employevenuefk FOREIGN KEY ( venueid )
  3   REFERENCES venue_sq_3530 ( venueid );
Table altered.

SQL>
SQL> ALTER TABLE " FEE_SQ_3530"
  2   ADD CONSTRAINT " FeeEmployeefk" FOREIGN KEY ( employeid )
  3   REFERENCES "EMPLOYEE_SQ_3530" ( employeid );
Table altered.

SQL>
SQL> ALTER TABLE " FEE_SQ_3530"
  2   ADD CONSTRAINT " FeeGuestfk" FOREIGN KEY ( guestid )
  3   REFERENCES guest_sq_3530 ( guestid );
Table altered.

SQL>
SQL> ALTER TABLE " FEE_SQ_3530"
  2   ADD CONSTRAINT " FeeServicefk" FOREIGN KEY ( serviceid )
  3   REFERENCES service_sq_3530 ( serviceid );
Table altered.

SQL>
SQL> ALTER TABLE " FEE_SQ_3530"
  2   ADD CONSTRAINT " FeeStayfk" FOREIGN KEY ( stayid )
  3   REFERENCES stay_sq_3530 ( stayid );
Table altered.

SQL>
SQL> ALTER TABLE " FEE_SQ_3530"
  2   ADD CONSTRAINT " Feevenuefk" FOREIGN KEY ( venueid )
  3   REFERENCES venue_sq_3530 ( venueid );
Table altered.

SQL>
SQL> ALTER TABLE stay_sq_3530
  2   ADD CONSTRAINT stayguestfk FOREIGN KEY ( guestid )
  3   REFERENCES guest_sq_3530 ( guestid );
Table altered.

SQL>
SQL> ALTER TABLE stay_sq_3530
  2   ADD CONSTRAINT stayvenuefk FOREIGN KEY ( venueid )
  3   REFERENCES venue_sq_3530 ( venueid );
Table altered.

```

```

SQL>
SQL>
SQL> -----
SQL> -- File created - Wednesday-October-12-2022
SQL> -----
SQL> REM INSERTING into ASSIGNS_1." FEE_SQ_3530"
SQL> SET DEFINE OFF;
SQL> REM INSERTING into ASSIGNS_1."EMPLOYEE_SQ_3530"
SQL> SET DEFINE OFF;
SQL> REM INSERTING into ASSIGNS_1.FEE_SQ_3530
SQL> SET DEFINE OFF;
SQL> REM INSERTING into ASSIGNS_1.GUEST_SQ_3530
SQL> SET DEFINE OFF;
SQL> REM INSERTING into ASSIGNS_1.SERVICE_SQ_3530
SQL> SET DEFINE OFF;
SQL> REM INSERTING into ASSIGNS_1.STAY_SQ_3530
SQL> SET DEFINE OFF;
SQL> REM INSERTING into ASSIGNS_1.VENUE_SQ_3530
SQL> SET DEFINE OFF;
SQL>

```

37

Add VENUE, EMPLOYEE, and STAY Data

You will now enter in data for the rest of your tables

i. Go to your [SQL Developer](#)

ii. Add and commit the following data to VENUE each with their own ID.

Van Whinkle Hotel
Easy Fingers Spa
King of Clubs Golf Course

iii. Add and commit the following data to EMPLOYEE each with their own ID. Associate each employee to the correct manager, guest, and venue. Leave the foreign key ID columns NULL if there is no such association. Do not leave any other columns blank.

Bob Smith is the manager of the hotel with 20 years' experience. He makes \$80,000 per year and lives in Windsor Ontario. His manager is Deena Donor.

Bob Smith is the desk clerk with only 5 years of experience at the hotel. He makes \$15 per hour and lives in Toronto Ontario. His manager is the other Bob Smith.

Tanya Duncan does maintenance and has been with the hotel for 10 years. She makes \$5,000 per month and lives in London Ontario. Her manager is Bob Smith.

Deena Donor is the operation manager for the hotel.

iv. Add and commit the following data to STAY each with their own ID. Associate each

stay to the correct guest and venue. Do not leave any columns blank.

Kristoff Kurn checked in last year by the desk clerk Bob Smith on Feb 3rd and checked out on Feb 17th

Billy Elliot checked in by the desk clerk Bob Smith last year on Jan 20th and checked out Jan 21st

Justin Hackman checked in by the manager Bob Smith Feb 2nd and checked out Feb 3rd; checked in Feb 8th and checked out Feb 9th; checked in Feb 17th and checked out Feb 18th; and checked in Feb 26th and checked out Feb 27th

Deena Donor checked in by the desk clerk Bob Smith on the 1st of each month and checked out on the 2nd of each month from January through to April

vi. Add one row in STAY showing you were a guest that stayed last year from January 1 to Jan 2.

vii. In the FEE table associate each fee to the correct employee, stay, and venue. Do not leave any columns blank.

38

Submit FEE Data Tab Screen Shot

Add below multiple screen shots of the FEE data using the table's sub-tab. To receive marks your screenshot must includes the table name and all the table's column data.

FEEID	FEEDescription	FEECHARGED	DATECHARGED	SERVICED	GUESTID	STAYID	EMPLOYEEID	VENUEID
1	1 Salira Qureshi	0.22-10-10	(null)	(null)	11	(null)	1	1
2	2 Deposit	-50.21-02-03	2	2	1	(null)	1	1
3	3 Room Rate	50.21-02-03	3	2	1	(null)	1	1
4	4 Room Rate	50.21-02-04	3	2	1	(null)	1	1
5	5 Room Rate	50.21-02-05	3	2	1	(null)	1	1
6	6 Room Rate	50.21-02-06	3	2	1	(null)	1	1
7	7 Room Rate	50.21-02-07	3	2	1	(null)	1	1
8	8 Room Rate	50.21-02-08	3	2	1	(null)	1	1
9	9 Room Rate	50.21-02-09	3	2	1	(null)	1	1
10	10 Room Rate	50.21-02-10	3	2	1	(null)	1	1
11	11 Room Rate	45.21-02-11	3	2	1	(null)	1	1
12	12 Room Rate	45.21-02-12	3	2	1	(null)	1	1
13	13 Room Rate	45.21-02-13	3	2	1	(null)	1	1
14	14 Room Rate	45.21-02-14	3	2	1	(null)	1	1
15	15 Room Rate	45.21-02-15	3	2	1	(null)	1	1
16	16 Room Rate	45.21-02-16	3	2	1	(null)	1	1
17	17 Room Rate	45.21-02-17	3	2	1	(null)	1	1
18	18 Show Damage Repair	25.21-02-17	4	2	1	(null)	1	1
19	19 Carpet Repair	30.21-02-17	5	2	1	(null)	1	1
20	20 Deposit	0.21-01-20	2	3	2	(null)	1	1
21	21 Room Rate	30.21-01-20	3	3	2	(null)	1	1
22	22 Deposit	-20.21-02-02	2	4	3	(null)	1	1
23	23 Room Rate	35.21-02-02	3	4	3	(null)	1	1
24	24 Room Rate	35.21-02-03	3	4	4	(null)	1	1
25	25 Room Rate	35.21-02-17	3	4	5	(null)	1	1
26	26 Room Rate	35.21-02-28	3	4	6	(null)	1	1
27	27 Deposit	0.21-01-01	2	5	7	4	1	1
28	28 Room Rate	0.21-01-01	3	5	7	4	1	1
29	29 Deposit	0.21-02-01	2	5	8	4	1	1
30	30 Room Rate	0.21-02-01	3	5	8	4	1	1
31	31 Deposit	0.21-03-01	2	5	9	4	1	1
32	32 Room Rate	0.21-03-01	3	5	9	4	1	1
33	33 Deposit	0.21-04-01	2	5	10	4	1	1
34	34 Room Rate	0.21-04-01	3	5	10	4	1	1

39

SELECT Statement

The SELECT statement is made of several clauses. Each clause is a command that manipulates the results of the data being selected and returned. The key to writing high

	<p>performance queries is to understand how the database actually runs the SELECT statement.</p> <p>Here's an example of the basic SELECT statement</p> <pre>SELECT FirstName, LastName FROM GUEST WHERE City = 'Ottawa' ORDER BY LastName ASC, FirstName ASC</pre> <p>The columns are selected, the filter is applied using the WHERE. Those filtered results are then sorted using ORDER BY.</p> <p>For this query involving GROUP BY</p> <pre>SELECT GuestID, SUM(FeeCharged) AS TotalFeeCharged FROM FEE WHERE FeeType = 'Damage' GROUP BY GuestID HAVING SUM(FEECharged) > 50 ORDER BY TotalFeeCharged DESC</pre> <p>The columns are GuestID and FeeCharged are selected but not shown. These results are then filtered by the WHERE. The resulting rows are then aggregated by GuestID and the SUM is calculated. The HAVING filter is applied to these SUMs. The ALIAS TotalFeeCharged is not known until when the ORDER BY is executed</p>
<p>40</p>	<h2>Any Blanks Query</h2> <p>Databases rarely have completely filled in data. Often when merging data from different sources you will need fill in the missing data. To quickly check what is missing, you would check if any column within a table's record is blank. A blank in a database is not an empty string. It is a NULL. A common error programmers make is using the equal operator (=) when checking for blanks. A blank is nothing, no value. So you can't use the equal operator for the check. You use the IS operator instead.</p> <ol style="list-style-type: none"> Go to your SQL Developer Run <code>INSERT INTO GUEST_NN_SSSS VALUES (9999);</code> where NN_SSSS is the first/last name initials with the last four digits of your student number Use the following pattern to check for blanks in your GUEST table (DO NOT use SELECT * FROM...) <pre>SELECT Field1, Field2, ... FieldN FROM TABLE_NAME WHERE Field1 IS NULL OR Field2 IS NULL OR... FieldN IS NULL;</pre> <ol style="list-style-type: none"> Add an ORDER BY clause to sort from Z to A on last names and from A to Z on first names. Run your SQL to confirm it works.

41

Submit IS NULL Screen Shot

Go to SQLPLUS and maximize your DOS window. Re-run your query. Copy and paste a screenshot of the query and its results below. To receive marks your screen shot must show tables having your initials and last four digits of your student number.

```

C:\Users\Saira>SQLPLUS
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Oct 12 22:51:47 2022
Version 19.0.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Enter user-name: ASSIGNS_1
Enter password:
Last Successful login time: Wed Oct 12 2022 22:51:16 -04:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.0.0.0.0

SQL> SELECT FirstName, LastName, GuestID, Province
2 FROM guest_sq_3530
3 WHERE FirstName IS NULL OR LastName IS NULL Or GuestID IS NULL Or Province IS NULL;

no rows selected

SQL> ORDERBY LastName DESC
SP2-0734: unknown command beginning "ORDERBY La..." - rest of line ignored.
SQL>
SQL> REM Saira Qureshi 3530_

```

42

Equal NULL

- Go to your [SQL Developer](#)
- Change each IS operator to = (equal) and run your query to confirm it works

43

Submit Equal NULL Screen Shot

Go to SQLPLUS and maximize your DOS window. Re-run your query. Copy and paste a screenshot of the query and its results below. To receive marks your screen shot must show tables having your initials and last four digits of your student number.

```

SQL> REM Saira Qureshi 3530
SQL>
SQL> SELECT FirstName, LastName, GuestID, Province
2 FROM guest_sq_3530
3 WHERE FirstName = NULL OR LastName = NULL Or GuestID = NULL Or Province = NULL;

no rows selected

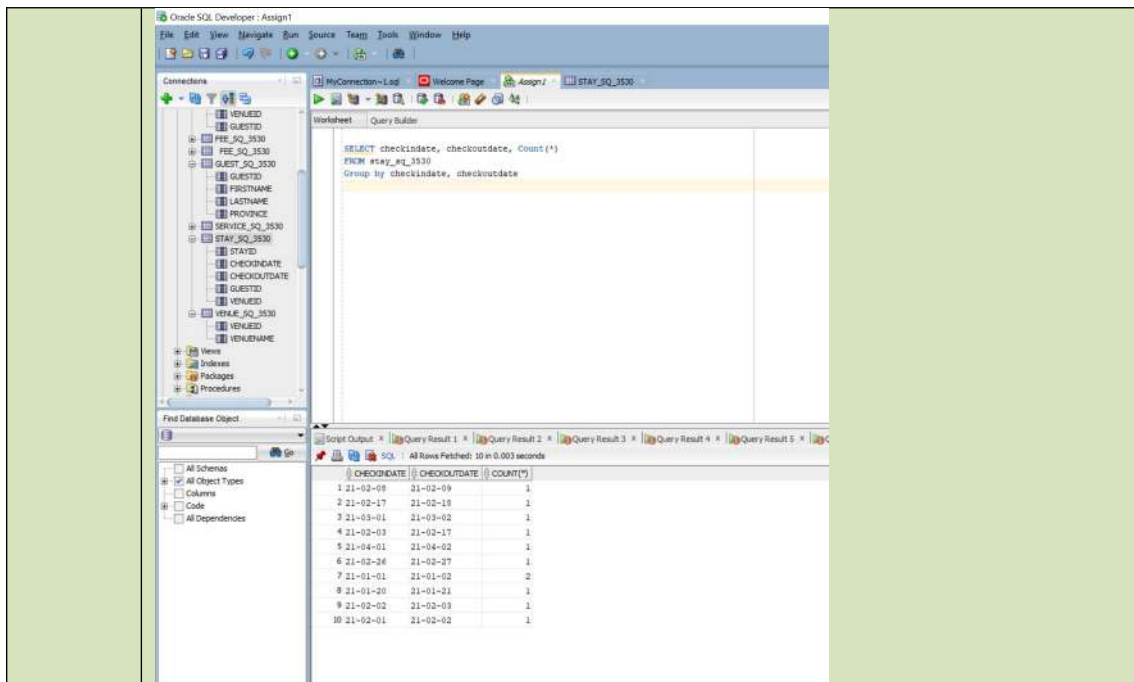
SQL> ORDERBY LastName DESC
SP2-0734: unknown command beginning "ORDERBY La..." - rest of line ignored.
SQL>
SQL>
SQL>

```

44

Number Of Days Per Stay Query

	<p>Management wants a report that compares which guests are staying for which number of days. This will allow management to target these guests for promotions and discounts.</p> <p>i. The first step is to create the days count calculation. Change the query below so it counts the number of days or times the room rate was charged for each stay. (Use StayID for the calculation not GuestID) Give the alias for count to be DayCount.</p> <pre>SELECT CustomerID, SUM(Amount) As TotalAmount FROM INVOICE WHERE InvoiceType = 'Construction' GROUP BY CustomerID</pre> <p>ii. Confirm your query runs correctly.</p> <p>iii. Move the query you wrote into a subquery having the alias TOTAL and join it to the guest table. Use the pattern below as a guide. Return the guest's first name, last name, and DayCount.</p> <pre>SELECT CustomerName, CustomerCity, TotalAmount FROM CUSTOMER JOIN (SELECT CustomerID, SUM(Amount) As TotalAmount FROM INVOICE GROUP BY CustomerID) TOTAL ON TOTAL.CustomerID = CUSTOMER.CustomerID</pre> <p>You will need to add the GuestID as a selected column in your subquery to complete the join</p> <p>iv. Confirm your query runs correctly</p> <p>v. Add a filter using WHERE at the end of your query to include only guests that stayed 1 day or 6 or more days. Confirm you query runs correctly.</p>
<p>45</p>	<p>Submit Number Of Day Stayed Screen Shot</p> <p>Go to SQLPLUS and maximize your DOS window. Re-run your query. Copy and paste a screenshot of the query and its results below. To receive marks your screen shot must show tables having your initials and last four digits of your student number.</p>



46

Expenses including Room Rate Query

Management wants a report that compares how much guests are spending. They want deposits excluded from the total calculation.

- Copy the Number Of Days Stayed query
- Change the query to return the guest's first name, last name, service received, the fee description, the fee charged, and the total charged for that guest for that stay. You will need to join on the GUEST table, FEE table, and TOTAL subquery using GuestID and StayID. Be careful when to join on GuestID to avoid duplicates. Confirm your query runs correctly.
- ADD a WHERE to both your subquery and main / top-level query to filter out service names that are deposits. Confirm your query runs correctly.
- Add an ORDER BY at the end of the top level query to sort the totals calculated from highest to lowest. Confirm your query runs correctly.
- Add a HAVING to your subquery to filter for totals that exceed \$40. Confirm your query runs correctly.

47

Submit Expenses including Room Rate Screen Shot

Go to SQLPLUS and maximize your DOS window. Re-run your query. Copy and paste a screenshot of the query and its results below. To receive marks your screen shot must show tables having your initials and last four digits of your student number.

	<paste screenshot below>
48	<p>Expenses excluding Room Rate Query</p> <p>Management wants a report that compares how much guests are spending. They want deposits and room rate fees excluded in the total calculation.</p> <ol style="list-style-type: none"> Copy the Expenses Including Room Rate query Change the query so that room rates are excluded from both the subquery calculation and from the top-level query rows returned. The Use NOT IN operator to accomplish this. Confirm your query runs correctly.
49	<p>Submit Expenses excluding Room Rate Screen Shot</p> <p>Go to SQLPLUS and maximize your DOS window. Re-run your query. Copy and paste a screenshot of the query and its results below. To receive marks your screen shot must show tables having your initials and last four digits of your student number.</p> <p><paste screenshot below></p>
50	<p>Generic Report Query</p> <p>Management wants to reuse reports based on specific customizations. This can be done using the IN operator and a new table in the database called REPORT_CONFIG.</p> <ol style="list-style-type: none"> Using SQL Developer create the REPORT_CONFIG table with the column ServiceName Add the rows "Room Rate" and "Room Deposit" Copy the Expenses Excluding Room Rate Query Change the NOT IN to IN for each WHERE clause Change the contents of each IN to a SELECT statement using the following as a guide <pre>SELECT FirstName FROM EMPLOYEE WHERE FirstName IN (SELECT NickName FROM MY_LIST)</pre> Confirm your query runs correctly.

<p>51</p>	<p>Submit Generic Report Screen Shot</p> <p>Go to SQLPLUS and maximize your DOS window. Re-run your query. Copy and paste a screenshot of the query and its results below. To receive marks your screen shot must show tables having your initials and last four digits of your student number.</p> <p><paste screenshot below></p>
<p>52</p>	<p>From Ottawa Query</p> <p>Management wants a list of employees and guests that reside in Ottawa so they can send out surveys. These surveys would help the hotel explore other venues and services.</p> <ol style="list-style-type: none"> Create a query that returns each employee's first name, last name, city, and title Copy the query and change it so it returns each guest's first name, last name, and city Combine the two queries together using a UNION Confirm the query runs successfully <p>Unions require the same number of columns for each subquery. You will need to add blank columns in one of your subqueries for your union to work</p>
<p>53</p>	<p>Submit From Ottawa Screen Shot</p> <p>Go to SQLPLUS and maximize your DOS window. Re-run your query. Copy and paste a screenshot of the query and its results below. To receive marks your screen shot must show tables having your initials and last four digits of your student number.</p> <p><paste screenshot below></p>
<p>54</p>	<p>Employees Query</p> <p>Management wants a list of employees and the name of the manager each employee reports to.</p> <ol style="list-style-type: none"> Create a query that returns each employee's first name, last name, city and their manager's first name, last name and city. Use the following query as a guide.

	<pre>SELECT SMALL_BAG.Colour, BIG_BAG.Color FROM SMALL_BAG LEFT JOIN SMALL_BAG BIG_BAG ON SMALL_BAG.BigBagID = BIG_BAG.BagID</pre> <p>vi. Confirm your query runs correctly.</p>
55	<h3>Submit Employees Screen Shot</h3> <p>Go to SQLPLUS and maximize your DOS window. Re-run your query. Copy and paste a screenshot of the query and its results below. To receive marks your screen shot must show tables having your initials and last four digits of your student number.</p> <p><paste screenshot below></p>
56	<h3>Employees are Guests Query</h3> <p>Management wants to see which of their employees are guests and heavy users of their employee discount. This information can be used for sales and/or for determining compensation packages.</p> <p>i. Copy the Employees query</p> <p>ii. Add a WHERE clause using the IN operator to filter for employees that are also guests</p> <p>You use IN and not JOIN because the IN operator returns only one column while JOIN returns more than one. This makes the IN operator not only easier to read but faster to execute than the JOIN.</p> <p>iii. Confirm your query runs correctly.</p>
57	<h3>Submit Employees are Guests Screen Shot</h3> <p>Go to SQLPLUS and maximize your DOS window. Re-run your query. Copy and paste a screenshot of the query and its results below. To receive marks your screen shot must show tables having your initials and last four digits of your student number.</p> <p><paste screenshot below></p>
58	

	<h2>Data Extract Query</h2> <p>Management wants to create a data extract of which employees are customers that have last names with Mc or Mac. From this information they will determine how to execute a St. Patricks marketing campaign.</p> <p>i. Copy the Expenses including Room Rates query. Remove all filters (WHERE clauses) from the query. Confirm the queries</p> <p>ii. Copy the Employees are Guests query</p> <p>vii. Merge the two queries together into a single query</p> <p>viii. Confirm your query runs correctly.</p> <p>iii. Add a WHERE clause to filter for the Guest's Last Name to filter for names that start with either 'Mac' or 'Mc'. You need to use UPPER since Oracle is case sensitive. Use the following pattern as a guide.</p> <pre>SELECT * FROM CUSTOMER WHERE UPPER(LastName) LIKE 'A';</pre>
59	<h2>Submit Data Extract Screen Shot</h2> <p>Go to SQLPLUS and maximize your DOS window. Re-run your query. Copy and paste a screenshot of the query and its results below. To receive marks your screen shot must show tables having your initials and last four digits of your student number.</p> <p><paste screenshot below></p>
60	<h2>End of Lab</h2>