IT 4153 Advanced Database

Module 1 SQL Review

|  |
| --- |
| **Introduction and Module Summary** |
| In this module, you will review database design and Structured Query Language (SQL). SQL is the standard language for relational database management systems. SQL knowledge is the prerequisite to this course. Depending on when you took an introduction to databases course, this module will take you anywhere from 6 to 12 hours of work. Spending enough time on this review will help you to complete other modules in this course. |
| **Objectives and Outcomes** |
| This module directly supports **highlighted** course outcome(s)  Students who complete this course successfully will be able to   1. Describe current and emerging database models and technologies; 2. **Develop functions and procedures for data manipulation and database access auditing;** 3. Describe database monitoring and performance tuning; 4. Describe database security and administration issues, including backup and recovery; 5. Explain the concepts of data warehousing and data mining   **Module outcomes and activities:**   |  |  |  | | --- | --- | --- | | After completing this module, students will be able: | to develop conceptual, logical and physical data models | to use SQL for data manipulation and data extraction | | Readings | introduced | introduced | | Lab 1.1 |  | reinforced | | Practice exercise (complete before lab 1.2) | reinforced | reinforced | | Lab 1.2 | reinforced | reinforced | | SQL quiz (complete after lab 1.2) | mastered | mastered | |  |  |  | |
| **Assigned Reading** |
| 1. SQL <http://docs.oracle.com/database/121/CNCPT/sqllangu.htm#CNCPT015> 2. Readings linked through the module. 3. Intro to SQL 1-10 <http://sqlcourse.com/intro.html> 4. Intro 2 to SQL 1-10 <http://sqlcourse2.com/> |
| **Optional Reading** |
| 1. Oracle Relational Data Structures <https://docs.oracle.com/cd/E11882_01/server.112/e40540/part_datstr.htm> 2. Oracle 12c <https://docs.oracle.com/database/121/nav/portal_4.htm> |
| **Assessments and Assignments** |
| 1. Lab 1.1 (10 points) 2. Lab 1.2 (10 points) 3. SQL Quiz (10 points) |
| **Topics** |
| Open the navigation pane |
| **Module Checklist** |
| This is the suggested order of the completion of this module.  Save a copy of this file on your computer and make notes in this document while you are completing your assignments. Use the table below to keep track of your progress.   |  |  | | --- | --- | | **Activity** | **Completion** | | Read this module and watch the recorded lecture (1 hour) | NO | | Read assigned readings (0 to 7 hours) | NO | | Complete practice exercise (1 hour) | NO | | Complete lab 1.1 (20 min) | NO | | SQL quiz (20 min) | NO | | Complete lab 1.2 (2 hours 20 min) | NO | | Complete Module feedback at the end of the module | NO | | Read feedback provided for your discussion and lab. | NO | |
|  |

# Database design

## Data Models

Database design is the process of producing a detailed data model of a database.

Data models:

* Conceptual Data Model: important entities and their highest-level relationships are defined
* Logical Data Model: all entities and relationships among them are defined; many-to-many relationships should be resolved; logical model includes for each entity
  + attributes
  + primary key
  + foreign key(s)
* Physical Data Model: different database management system may cause the physical data model to be quite different from the logical data model; physical data model includes all objects (indexes, constraints, partitions, clusters, etc) required to create relationships between tables or achieve performance goals. Order of columns is also important for good performance.

Some definitions

* Rows and tables <http://www.tomjewett.com/dbdesign/dbdesign.php?page=tables.php>
* Associations <http://www.tomjewett.com/dbdesign/dbdesign.php?page=association.php>
* Keys <http://www.tomjewett.com/dbdesign/dbdesign.php?page=keys.php>

## Normalization and Denormalization

Database normalizing is one of the ways to decrease storage requirements and increase data integrity. There are several rules for database normalization. Each rule is called a "normal form." Information about three normal forms and normalization example can be found at <http://technet.microsoft.com/en-us/library/cc505842.aspx>

Sometimes to capture history or optimize performance of a database driven application we have to use denormalization. You have to understand that denormalization might cause data integrity problems that have to be addressed in application design. Later in this course you will learn how and when to use pre-joined tables, report tables, mirror tables, split tables, combined tables and other types of denormalization. In this module you need to understand benefits and drawbacks of denormalization <http://technet.microsoft.com/en-us/library/cc505841.aspx>

## Data Integrity

Some definitions:

* Entity integrity: each row of a table has a unique identifier.
* Semantic integrity: the data in the columns properly reflects the types of information the column was designed to hold. Example: gender field can contain F, M, or NULL (if nulls are allowed).
* Referential integrity: the relationships between tables are enforced.
* DBMS Integrity: index consistency, pointer consistency (for objects stored outside of database file), backup consistency

Entity & Referential Integrity <https://docs.oracle.com/javase/tutorial/jdbc/overview/database.html>

## Database Objects

Each database consists of one or more logical storage units called tablespaces. Tablespaces contain indexes, views and table data.

Each tablespace consists of one or more files called datafiles.

A schema is a collection of database objects.

In Oracle schema is owned by a database user and has the same name as that user. It is possible to transfer objects from one schema to another or do a user-level export and then import.

A table consists of a defined number of columns and any number of rows.

Each table may have multiple indexes. Indexes may provide a faster way to access table data.

Each table may have multiple views. Views may be associated with more than one base table.



 tablespace

## Structured Query Language (SQL)

SQL was standardized first by the ANSI and then by the ISO. It consists of

* data definition language (DDL) Examples: CREATE, ALTER, DROP statements
* data manipulation language (DML) Examples: SELECT, UPDATE, INSERT statements
* Data Control Language (DCL) Examples: GRANT, REVOKE statements
* Transaction Control (TCL) Examples: COMMIT, ROLLBACK statements

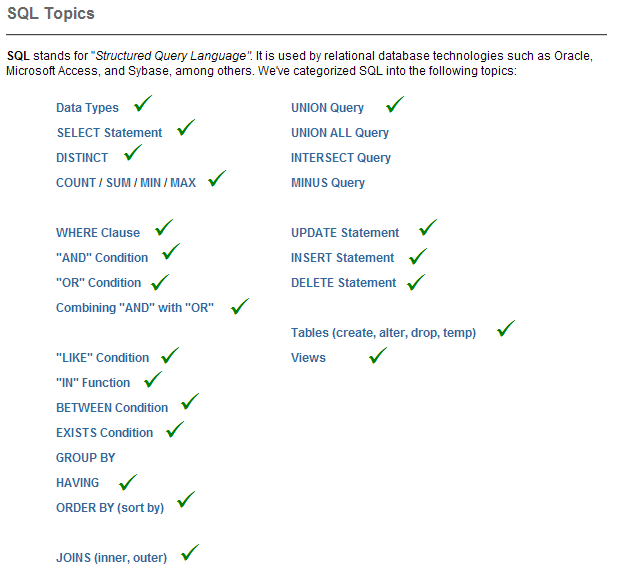
We will review only DML and DDL.

Data Definition Language (DDL) statements let you to perform these tasks:

* Create, alter, and drop schema objects
* Grant and revoke privileges and roles
* Analyze information on a table, index, or cluster
* Establish auditing options
* Add comments to the data dictionary

Data manipulation language (DML) statements access and manipulate data in existing schema objects.

Use <http://www.techonthenet.com/sql/index.php> You should know everything marked with green check mark ✓. I encourage you to read about views also.



Understand how NULL is used in SQL <http://en.wikipedia.org/wiki/Null_%28SQL%29#Data_Manipulation_Language>

YOU HAVE TO COMPLETE ALL LABS YOURSELF IN ORDER TO PASS THE TEST. ALL TEST QUESTIONS WILL REQUIRE YOU TO WRITE CODE.

## Indexes

Read an introduction from <https://www.tutorialcup.com/dbms/database-index.htm> We will discuss type of indexes later, just read about basic indexes.

## HAVING Clause

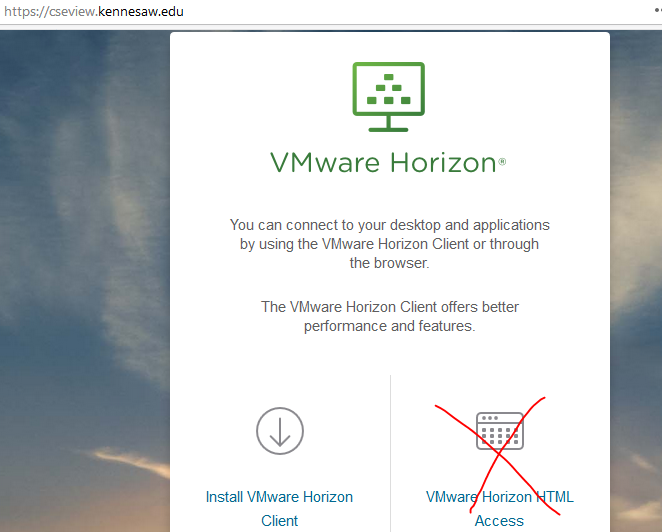
To specifies a search condition for a group or an aggregate a HAVING clause can be used <http://www.w3schools.com/sql/sql_having.asp>

# 

# Lab 1.1 SQL Review

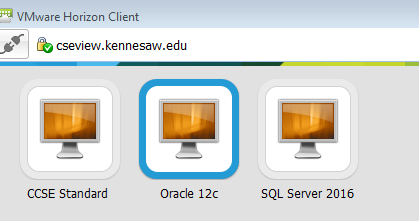
**Pre-Lab Oracle**

Note: We will use Oracle 12c VM for all labs in this class. Go to <https://cseview.kennesaw.edu/> Install VMWare Horizon Client, do not use HTML client.



Quick step-by-step guide is available at <http://ccse.kennesaw.edu/docs/CCSE%20Virtual%20Desktop%20Login%20Instructions.pdf>

After you login, you should see either Oracle 12c or IT4153 VM



The VMs should be persistent and all your changes should be saved, but you should **always save your code** in case something happens to the VM. I suggest you to have a local copy of all your code.

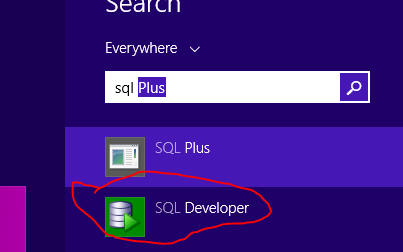
You can write all your code on the host machine and then paste it in the VM or save your files on OwlDrive (u:\ drive) in the vm <https://apps.kennesaw.edu/files/pr_app_uni_cdoc/doc/Using_the_OwlDrive_in_Windows.pdf>

You can also use an USB drive or google docs inside the VM or shared folders from your host machine.

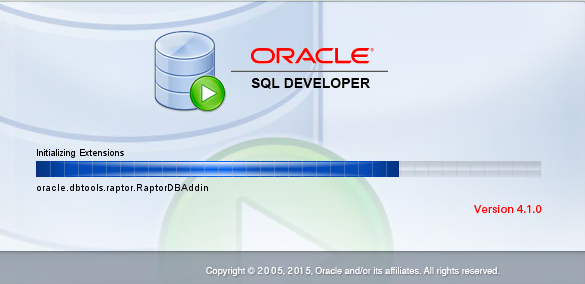
Unfortunately, now you cannot copy from VM and paste to the host. I hope this issue will be solved soon.

Use Oracle 12c to complete this lab. As alternative, you can create your own Oracle 12c VM (see <http://ksuweb.kennesaw.edu/~speltsve/files/sql.php> )

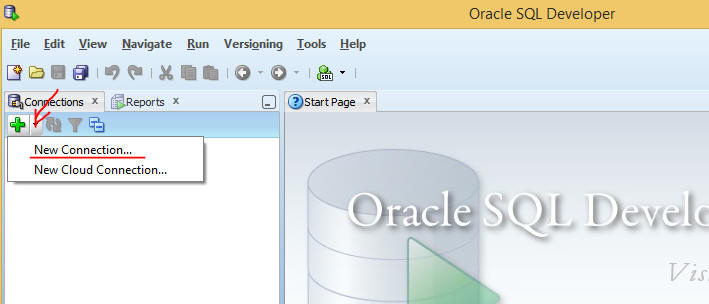
Start Oracle VM and Start SQL Developer

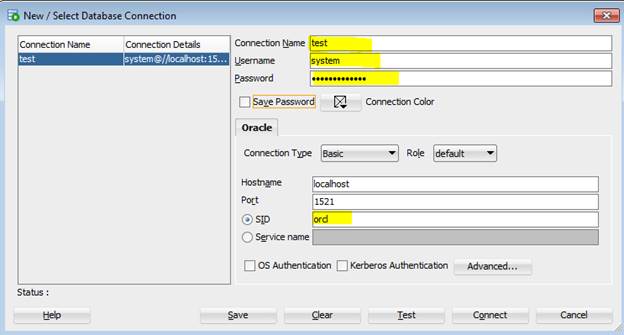


Start SQL Developer (Note: ignore .net error)



Create new connection

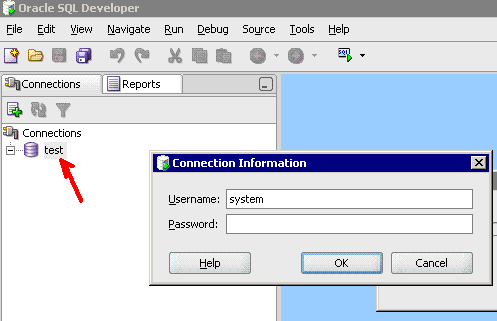




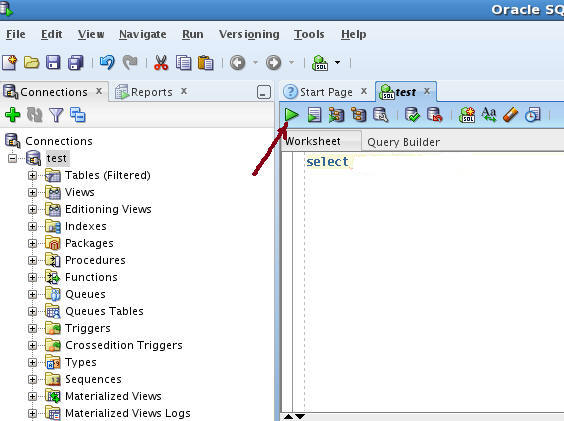
Create new **Database Connection**

connection name: **test**username: **system**password: **Oracle\_admin01 (Capital O in Oracle and zero one at the end)**sid: **orcl (the last letter is L)**

Test the connection and if the status is a success, then click save, then connect.



Double click the connection, it will open new worksheet and your screen should be similar to one below



Execute SELECT \* FROM DBA\_TABLES;

To execute SQL statement click GREEN ARROW

sql

To execute several statements or PL/SQL code click RUNS SCRIPT BUTTON (next to Execute Statement button)

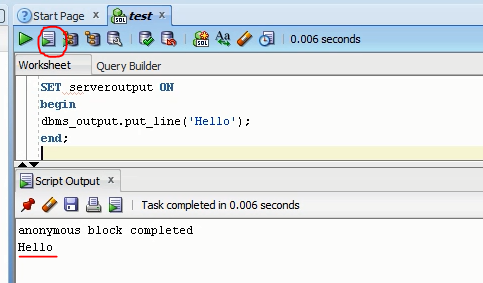
SET serveroutput ON

begin

dbms\_output.put\_line('Hello');

end;

1. Open a new SQL worksheet.
2. Type the code. The first line turns on dbms\_output. The third line displays "Hello".
3. Click Run Script button.
4. Output will be displayed in Script Output window.

Now you have working Oracle environment for this class.

**Lab Instructions**

Create lab report file. Use SQL Developer to run code below.

1. Insert a screenshot that shows results of execution of the following code (2 point for the screenshot)

SELECT \* from dba\_users;

1. Insert a screenshot that shows results of execution of the following code (2 point for the screenshot). SYSDATE returns current day and SYS\_CONTEXT('USERENV','OS\_USER') returns operating system user running the session.

SET serveroutput ON

begin

dbms\_output.put\_line(SYSDATE || ' ' || SYS\_CONTEXT('USERENV','OS\_USER'));

end;

1. Post a question or provide an answer on the discussion board “Help with labs REQUIRED weekly posts”. Provide a screenshot showing your post. (1 point)
2. Create table ***myTable*** with two column, insert at least one row and select data you inserted. (5 points for create/insert/select code and the screenshot showing results)
3. Complete practice exercise linked to the module. You do not need to submit it for grading.

**Feedback:**

Difficulty (-2 - too easy ... 0 - just right ... 2 - too hard)

Interest level (-2 - low interest ... 0 - just right ... 2 - high interest)

Time to complete (min)

Make a suggestion to improve

**What to submit:**

**ONE** report file that includes answers to the above questions. Do not include module instructions or prelab. Provide code and screenshots of successful execution where applicable. If I cannot copy your code CREATE TABLE, INSERT, etc from your report and run it, you will not receive points.  
NOTE: All work turned in for this class must meet the style and submission guidelines [http://ksuweb.kennesaw.edu/~speltsve/files/style\_and\_submission\_guide\_d2l.pdf](http://ksuweb.kennesaw.edu/%7Espeltsve/files/style_and_submission_guide_d2l.pdf) Work that does not meet the style and submission criteria will not be graded.

# Lab 1.2 Instructions

(1 point for each question)

Suppose you have two tables: Table Product:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PROD\_ID | PROD\_NAME | PROD\_PRICE | PROD\_PROD\_DATE | PROD\_VENDOR |
| 1101 | Table | 100 | 1-Jan-18 | 2 |
| 1102 | Chair | 80 | 3-Mar-18 | 3 |
| 1103 | Armchair | 90 | 6-May-18 | 2 |
| 1104 | Nightstand | 110 | 4-Apr-18 | 1 |
| 1105 | Bed | 200 | 3-Mar-18 | 3 |
| 1106 | Dresser | 150 | 5-May-18 | 3 |
| 1107 | Daybed | 190 | 10-Feb-18 | 2 |

Table Vendor:

|  |  |  |
| --- | --- | --- |
| VEND\_ID | VEND\_NAME | VEND\_ST |
| 1 | Green Way Inc | GA |
| 2 | Forrest LLC | NC |
| 3 | AmeriMart | NC |

1. Post a question or provide an answer on the discussion board “Help with labs REQUIRED weekly posts”. Provide a screenshot showing your post.
2. Identify primary and foreign keys. Draw the relational schema to show the relationship between two tables using Crow Foot Notation <https://www.codeproject.com/Articles/878359/Data-modelling-using-ERD-with-Crow-Foot-Notation> You can use any software
3. Create tables. Do not forget about primary and foreign keys. Provide SQL code and screenshot of DESC statements.
4. Insert data. Provide SQL code and screenshot of select statements.
5. Suppose you wanted quick lookup capability to get a listing of all products supplied by a given vendor. Which table would be the basis for the INDEX table, and what would be the index key? Create the index. Provide SQL code.
6. Create a view that lists all products in alphabetical order and names of their vendors. Select all rows from the view and provide a screenshot. Hint: Use join.
7. Find all products that were manufactured between February 15 and April 15, 2018. Provide SQL code and the screenshot of the result.
8. Delete vendor AmeriMart form Vendor table. Provide SQL code and the result of select statement for both tables before deletion and after deletion. If you still have bed, dresser and chair in your product table, go back to #3 and fix foreign keys.
9. Update table Products. Change the name of #1101 to Cherry Table.
10. Find all vendors who supplied less than two products. Provide SQL code and the screenshot of the results.

**Feedback:**

Difficulty (-2 - too easy ... 0 - just right ... 2 - too hard)

Interest level (-2 - low interest ... 0 - just right ... 2 - high interest)

Time to complete (min)

Make a suggestion to improve

**What to submit:**

**ONE** report file that includes answers to above questions. Provide code and screenshots of successful execution where applicable. I should be able to copy your code and run it to test your solution.  
NOTE: All work turned in for this class must meet the style and submission guidelines [http://ksuweb.kennesaw.edu/~speltsve/files/style\_and\_submission\_guide\_d2l.pdf](http://ksuweb.kennesaw.edu/%7Espeltsve/files/style_and_submission_guide_d2l.pdf) Work that does not meet the style and submission criteria will not be graded.