Note: Reading is ordered chronologically from the Canvas modules. Linux commands and other syntax is not included because if we’re tested on this I’m going to be very surprised.

* Aside: [Here](https://www.youtube.com/playlist?list=PL_c9BZzLwBRK0Pc28IdvPQizD2mJlgoID) is a playlist I found that talks about databases that I found helpful.

## **DB Reading Packet 1 - DB Processing and Development**

Summary: Outlines the basic definitions of a database, DBMS, and DB processing system, as well as why they are favored over file-processing systems.

Review Questions:

* What is a database?
* What is a DBMS, and why is Excel *not* one? What features should a DBMS have?
* List the disadvantages of using a file-processing system.
* List the advantages of using a DB processing system. Also identify disadvantages.
* What does it mean to be a well-designed database?

Concepts/Definitions to know:

* Database, relational database
* Database management system (DMBS)
* Database application
* File-processing system
* Data integrity
* Query

## Textbook Chapter 1 - DBs and DB Users

Summary: Databases hold a collection of data, and represent something in the real world. They can be used for a variety of purposes depending on the type of user, with views supporting what actions a type of user can perform. These databases are implemented and maintained through DBMS software, usually by a database administrator and their database designers. Although databases have many advantages over file-processing systems, overhead costs may make its implementation less favorable.

Review Questions:

* What does CRUD stand for in relation to databases?
* What are the four main types of users? Give examples.
* What should a DBMS be able to do when maintaining/implementing a database?
* Why is the concept of canned transactions important for data abstraction/hiding?

Concepts/Definitions to know:

* Miniworld
* Application program
* Program-data independence
* Program-operation independence
* Data abstraction
* Views
* Concurrency control
* Transactions, isolation, and atomicity; canned transactions
* Database Admin (DBA)
* Database designer
* Business rules

## SQL Reading Packet 1- Intro to Oracle SQL

Summary: This reading packet outlines the most basic commands for use in Oracle SQL. It outlines spooling, creating/dropping tables, inserting rows into a table, starting an SQL script, selecting a whole table, and describing a table.

Review Questions:

* What are the different data types available for use in SQL\*Plus?
* Why do you need to drop a table before creation?
* What does cascade constraints do, and why is it important when dropping tables?
* Know the syntax for spooling, create, insert, and select.

Concepts/Definitions to know:

* SQL script
* Data definition language (DDL)
* Data manipulation language (DML)
* Data control language (DCL)

## DB Reading Packet 2 - More DB Fundamentals

Summary: This packet summarizes the different ways to express a relation and the four main elements of a database. It also touches on the history and reasoning behind the concept of relational databases, and includes examples of well- and poorly-structured relations.

Review Questions:

* Should be able to distinguish tabular, relation structure, and create table forms.
* What are the pros/cons of each depiction used to represent a relation?
* When is data duplication necessary/acceptable?

Concepts/Definitions to know:

* User data
* Metadata
* Indexes
* Application metadata

## Textbook Chapter 5 - Relational Data Model, rDB Constraints

Summary: This chapter introduces the concept of relational databases and the relational data model, as well as the mathematical definitions that are associated with it. It also discusses the idea of constraints and keys.

Review Questions:

* Why does a primary key have to be a candidate key?
* Why should NULL values be used sparingly? What can they represent?
* What does a relation being a set say about the characteristics of tuples stored in it?
* What are some examples of the different types of keys listed in this chapter?

Concepts/Definitions to know:

* Domain
* Relation schema, degree, and state
* Attribute
* Cardinality
* Current relation state
* NULL
* Superkey, key, minimal superkey, candidate key, primary key, foreign key, unique key
* Relational database schema
* Relational database state
* Integrity constraints
* Entity integrity constraint
* Referential integrity constraint
* Transaction

## DB Reading Packet 3 - Relational Model

Summary: This reading packet is very similar to textbook chapter 5 through also talking about the characteristics of the relational model, keys, and constraints. This packet also talks about the different types of relational operations you can perform as queries in SQL.

Review Questions:

* What are the constraints placed on a cell in a relation/table?
* How might the physical and semantic definitions of an attribute’s domain differ? Can you think of an example?
* Be able to write/identify functional dependencies given a relation and its attributes.
* A relation can have many minimal superkeys (AKA candidate keys) that can turn into primary keys. What should you consider when choosing the primary key for a relation?

Concepts/Definitions to know:

* Relation (same as relation state)
* relation = table = file
* tuple = row = record
* attribute = column = field
* Cell
* Functional dependency (and its notation)
* Relational algebra
* Relational operations: selection, projection, Cartesian product, equi-join, natural join

## SQL Reading Packet 2 - Writing Relational Operations

Summary: This reading packet summarizes the ways you can write relational operations. It also outlines common pitfalls when writing said operations, from omitting DISTINCT to using the wrong case in a conditional statement involving a string.

Review Questions:

* In what situations would a DBMS not create a “pure” relational projection? How do we ensure that it does?
* What is the syntax in an create-table statement to designate a foreign key?
* How do you insert a row where you only fill in a partial amount of its attributes?
* When is SQL case-sensitive and case-insensitive?
* Should be able to write and identify selection, projection, Cartesian product, equi-join, and natural join SELECT statements.
* Should be able to know when to specify the table name in a SELECT statement (ex: empl.salary), and that this should be consistent throughout the entire statement.

Concepts/Definitions to know:

* DISTINCT keyword
* SELECT, FROM, and WHERE clauses (and the course style standard)
* Join condition

## DB Reading Packet 4 - ER Modeling pt. 1

Summary: This reading packet introduces the concept of ER modeling, as well as the most basic features that should be included in an ER diagram. It also introduces the ideas of entities, relationships, and cardinalities.

Review Questions:

* What is data modeling, and when should you do it?
* Who is the person that determines the cardinalities, relationships, etc. that are important to model in the ER diagram?
* Should be able to create an ER diagram using the course style standard.

Concepts/Definitions to know:

* Entities, entity class
* Relationships, relationship class, relationship degree
* Attributes (simple, composite, single-valued, multi-valued)
* Identifiers
* Maximum and minimum cardinalities (1:1, 1:N, N:M)

## Textbook Chapter 3 - Data Modeling Using ER Model

Summary: This chapter of the textbook introduced ER modeling, the real-life process of creating a model, and the different scenarios that a model may depict. This chapter also includes two different examples of ER diagrams, but it does not use the course style standard.

Review Questions:

* Come up with an example of a composite attribute.
* Come up with an example of a multi-valued attribute.
* Come up with an example of a derived attribute.
* Come up with an example of a recursive relationship.
* What are the two structural constraints of a relationship type?
* What is an example of a partial key in relation to weak and strong entity types?

Concepts/Definitions to know:

* Conceptual, logical, and physical design
* Derived and stored attributes
* Complex attributes
* Key/Uniqueness constraint
* Value set/domain
* Participation constraint: partial, and total
* Role name
* Recursive relationship
* Weak and strong entity types
* Partial key

## DB Reading Packet 5 - ER Modeling pt. 2

Summary: This packet deliberates on how to model recursive relationships, weak and parent entity classes, and subtype and supertype entity classes. Examples of each, as well as the course style standard, are outlined in this packet.

Review Questions:

* Come up with examples for disjoint, overlapping, and union supertype/subtype classes.
* Where should you decide to place attributes when dealing with supertypes and subtypes (ex. in the subtype, or in the supertype, or both)?
* Be able to diagram subtypes and supertypes in the course style standard, as well as interpret them into plain English.

Concepts/Definitions to know:

* Supertype entities
* Subtype entities
* IS-A relationship
* Disjoint, overlapping, and union

## SQL Reading Packet 3 - WHERE and Functions

Summary: This packet delved deeper into the different operators and predicates available for use in SELECT statements. These can allow for a very large amount of possible queries to be created with as many tables as desired. This packet also introduces the concept of aggregate functions, like min() and max().

Review Questions:

* When are table names required before a column name in a SELECT statement?
* In general, be able to write SELECT statements using the operators and predicates listed below.
* What is the rule of thumb for the number of join conditions, given X amount of tables to join?
* What is the difference between count(<attr>) and count(\*)?

Concepts/Definitions to know:

* is NULL, is not NULL
* Boolean operators AND, OR, NOT, !=, <>
* BETWEEN operator
* LIKE operator; % and \_
* Column computations
* Column and table aliases
* IN predicate
* avg(), min(), max(), sum(), count()

## SQL Reading Packet 4 - Subselects, etc

Summary: This reading packet introduces the concept of subselects, which are essentially nested SELECT statements. Common errors in creating subselects is also discussed. It also details the concepts of projecting literals, concatenation, and the EXISTS predicate. Many examples for all of these concepts are in the packet, as well as the course style standards.

Review Questions:

* What are your different options in debugging an SQL script?
* Should be able to write and interpret the meaning of SELECT statements that contain subselects (possibly with more than two levels of nesting).
* What are the uses for projecting string literals?
* What is the notation for concatenating values? What can you use this for?
* Why is selecting a literal value in the correlated subquery preferred (and part of the course style standard)?
* Should be able to write and interpret the meaning of SELECT statements that contain an EXISTS predicate.
* What makes a correlated subquery and correlated condition different from normal subqueries and conditions?

Concepts/Definitions to know:

* Echo
* Subselects/Nested Select
* Concatenation
* EXISTS
* Correlated subquery
* Correlated condition