

CS 325 - Exam 1 Review Suggestions - Fall 2021

last modified: 2021-10-04

- You can receive (a maximum) ***5 POINTS BONUS*** on Exam 1 if you do the following:
 - Make a **hand-written** Exam 1 study sheet.
 - Submit a photo or scan of it saved as a .pdf, .png, .jpg, or .tiff to Canvas **by 9:00 am on Monday, October 11** such that I can read at least some significant CS 325 Exam 1 material on it
 - Please let me know if you have any questions about this, and I hope it helps you in reviewing course concepts more effectively before Exam 1.
 - You are **encouraged** to have this available as you are taking Exam 1.
- Exam 1 will be given in Canvas.
 - It will be **available from 9:00 - 10:30 am on Monday, October 11.**
 - It will be set up such that **you may only attempt Exam 1 once.**
 - That said, remember that **I will be available in the regular Zoom CS 325 session from 9:00 - 10:30 am on Monday, October 11.**
 - This will be set up so all microphones are muted, and you will only be able to chat with me.
 - **You are encouraged to LET ME KNOW about technical difficulties, so we can make provisions as needed!**
 - **And, I will have a small displayed window in the Zoom session with any typos or announcements that have come up so far during Exam 1.**
 - Exam 1 will be set up such that **you will be shown one question at a time,**
BUT there will be a list of question-links on the right-hand-side of the Canvas screen, and you can go back and forth between questions during that **one** exam attempt.
 - You are expected to work **individually** on the exam -- it is not acceptable during the exam to discuss anything on the exam with anyone else.
 - You may look up information from your Exam 1 study sheet, on-line, or from the course textbook during the exam, but note that if you take too long looking up material, you may have trouble completing the exam during the time period.
 - I expect there will be a few multiple-choice questions, and the rest will be short-answer questions.
 - BUT given the Canvas quiz question options, for those short-answer questions, you will be shown a larger area for your answer than you will need!
 - You will be reading and writing SQL statements and SQL*Plus commands.
 - You will be answering questions about concepts as well.
- A link to a packet of references and additional instructions - intended for use with Exam 1 - will be posted from 9:00 to 10:30 am on the course Canvas site, linked from the Exam 1 Instructions.
 - So, you can have it open in another browser window while you are taking Exam 1.

- This is intended both for reference and for use directly in some exam questions.
- You are responsible for material covered in class sessions, lab exercises, and homeworks; but, here's a quick overview of especially important material.
- You are responsible for the material covered through the end of Week 6 (Friday, October 1st), and through and including the Week 6 Lab Exercise and Homework 5.
- Your studying should include careful study of posted examples and notes as well as the lab exercises and homeworks thus far.
- Note that you are also responsible for knowing -- and following -- the course SQL style standards and the course ERD notation.

Introduction to DBMSs

- What is a DBMS? What is an example of a DBMS? What are some typical capabilities of a DBMS?
- What is a database? What is a relational database?
 - What are the 4 main elements of a database? (user data, metadata, indexes, & application metadata)
- What are some of the important limitations of file-processing systems?
 - What are some of the advantages of the database approach over these file-processing systems?
- What is metadata?
- What is a relation?
 - How do you represent a relation in relation structure form? ...in tabular form? ...in SQL `create table` statement form?
 - (possible question: here's a table/relation. Write a relation structure for it.)
 - What are the important "restrictions"/features of a relation?
 - (possible question: is the following a relation? Why not?)
 - What is a primary key? How do you indicate a relation's primary key in relation structure form?
- DBMS - database management system
 - What are some typical capabilities of a DBMS?
 - What is meant by DDL, DML, and DCL? What does each of these do?
 - What are some capabilities that a (high-end) DBMS might provide?
- You should know that a database design/schema defines a database's structure, and typically includes:
 - its tables,
 - relationships,
 - domains, and
 - business rules
- What are business rules?

- How do you create tables in Oracle SQL? How do you define relationships between tables?

Introduction to the relational model and relational operations

- Who developed it? When? (Codd, 1970) Why was it first resisted?
- What is a relation? (including: What are the restrictions?)
 - Be comfortable with relation/table, tuple/row/record, attribute/column/field terminology
 - Single-valued cells, no duplicate rows, order of rows/columns not important, column entries all of same "kind"/from the same domain; must have a primary key

Functional Dependencies and Key Definitions

- What is a functional dependency? What does it mean for one attribute to be functionally dependent on another?
 - Understand the \rightarrow notation;
 - What is the determinant in $A \rightarrow B$?
 - Given a relation in relation structure form, what are some functional dependencies that you can assume?
 - Does a determinant have to be a primary key? Why or why not?
 - Does a primary key have to be a determinant? Why or why not?
 - if $(A, B) \twoheadrightarrow C$, does it logically follow that $A \twoheadrightarrow C$ and $B \twoheadrightarrow C$?
 - if $A \twoheadrightarrow (B, C)$, does it logically follow that $A \twoheadrightarrow B$ and $A \twoheadrightarrow C$?
- What is a superkey? ...a minimal key? ...a candidate key? ...a primary key?
 - How does one indicate a primary key in a relation structure?
 - How does one indicate a primary key in a SQL `create table` statement?
 - How many attributes may be in a primary key?

Relational Operations

- The set-theoretic relational operations include union, difference, intersection, Cartesian product -- we have discussed only Cartesian product thus far;
- The relation-theoretic relational operations include rename, selection, projection, equi-join/natural join/other joins, and division -- we have discussed only selection, projection, equi-join, and natural join so far;
- At this point, you are expected to know and understand selection, projection, equi-join, natural join, and Cartesian product;
 - Expect to have to show that you can perform some or all of these operations on example relations; (note that I may describe the desired relational operation in words OR give it in SQL form);
 - What is an equi-join? How does it differ from a natural join?

- Know the `relation_name.attribute_name` notation;
- You should be able to express queries as combinations of relational algebra operations.

Introduction to the Entity-Relationship Model

- What is (should be) the database development process? What is a database model? What are some of the general strategies for developing a database model?
 - Important: the idea is that you come up with a database model *before* you come up with tables! Why should you develop a data model/database model before starting to create database tables?
 - Note: there is more than one type of database model; we are focusing on the most commonly-used, the entity-relationship model
- Remember: you are responsible for the entity-relationship diagram (ERD) notation given in class and in the course handouts; see also the guidelines mentioned in the Homework 5 handout.
- What are the elements of the E-R model? (entities, attributes, identifying attributes, relationships, cardinalities)
 - (Note that "entity" is often used interchangeably for both entity class and entity instance; if you are unsure whether I mean entity class or entity instance in an exam question, please ask me during the exam!)
- What is an entity class? How is it depicted in an ERD? What is/are an entity class's identifying attribute(s)?
 - Remember: an entity class is *not* equivalent to a table or relation!
 - (Eventually, each entity class will *result in one or more* corresponding tables/relations in the database schema/design that we develop from a model;)
- What is a relationship? How is it depicted in an ERD?
- What is an attribute? What is its domain? How is an attribute depicted in an ERD (according to our course ERD standards)?
 - Make sure it is clear to you what is *not* an attribute in an ERD as well: attribute lists in an ERD should include *no* relationship-related information. *Only* the relationship lines in the ERD show the relationships between entity classes!
- What are maximum cardinalities of a relationship?
 - What are the possible maximum cardinality values (typically)? (one and many)
 - Based on the maximum cardinalities, what are the 3 (4) "kinds" of relationships? (1:1, 1:N, N:M)
 - Given the pertinent information about a scenario, you should be able to determine which maximum cardinalities are appropriate for a relationship;
 - According to our course ERD standards, how are maximum cardinalities depicted in an ERD? You should be able to read and understand an ERD's maximum cardinalities, and you should be able to create an ERD with appropriate maximum cardinalities;
- What are minimum cardinalities of a relationship?

- What are the possible minimum cardinality values (typically)? (0 and 1)
- Given the pertinent information about a scenario, you should be able to determine which minimum cardinalities are appropriate for a relationship;
- According to our course ERD standards, how are minimum cardinalities depicted in an ERD? You should be able to read and understand an ERD's minimum cardinalities, and you should be able to create an ERD with appropriate minimum cardinalities;
- What is a supertype entity class? What is a subtype entity class?
 - How is a supertype/subtype relationship depicted in an ER diagram? (Remember to follow class style standards for these)
 - What is meant by having a `d` in the circle in depicting supertype/subtypes entity classes? ...having an `o` in that circle? ...having a `u` in that circle?
- It is **very likely** that you will be asked questions about a given ERD, to see if you can read it correctly;
- It is **very likely** that you will be asked to either draw or complete an ERD given scenario information;

Basics of Oracle SQL and Oracle SQL*Plus

- You will be required to read *and* write proper syntax SQL and SQL*Plus statements;
 - (By "read", I mean that I may give you a statement and ask you questions about it; I could also give you various table contents, and ask you what the results of running a given statement would be;)
 - And, of course, I could ask you to write a SQL statement that would perform a specified action or query;
- How do you start up SQL*Plus on nrs-projects?
- How do you create a table using SQL?
 - How do you define attributes? What are some of the common data types?
 - Be comfortable with further ways you can restrict the domains of attributes (`not null`, `default`, `check`)
 - How do you define a table's primary key in SQL?
 - What is a foreign key? How do you define a foreign key in SQL?
 - What kind of integrity checking do you get "automatically" in Oracle when you make an attribute a foreign key? ...a primary key?
- How do you insert rows into a table? (know both variants)
 - Which version of `insert` do you need to use to make sure you get any `default` values for attributes that have them?
- How do you use a `select` statement to show the contents of a table?
- What is a SQL script? How is it created? How is it run?
 - How can you write a SQL comment?
- You should be familiar with the SQL*Plus commands we have discussed so far, especially:

- Which command can be used to list the column definitions for a table? (*describe*)
- Which can be used to start spooling results to a file (and which can stop such spooling)?
- Which can be used to execute a SQL script?
- Which can be used to output specified characters to the screen?
- How can you delete a table?
- What is the basic syntax and basic semantics of the SQL `select` statement?

Writing "pure" relational operations using a SQL `select` statement

- How can you write a (pure) relational **projection** using a SQL `select` statement?
 - What SQL keyword can mean the difference between a "pure" relational projection and a not-"pure" one? What is the effect of this keyword? Where must it be placed?
- How can you write a relational **selection** using a SQL `select` statement?
- How can you write a relational **Cartesian product** using a SQL `select` statement?
- How can you write a relational **equi-join** using a SQL `select` statement? a relational **natural join**?
- How can these relational operations be combined within a single SQL `select` statement?
 - How can you express a query/question as a **combination** of relational algebra operations?
 - How can you express a query/question as a SQL `select` statement?

More on the basic SQL `select` statement

- Be familiar with the `where` clause possibilities discussed so far:
 - = < > <= >= <> !=
 - in
 - is null, is not null
 - and, or, not
 - between
 - like, %, _ (underscore)
- aliases
 - What is a table alias (within the `from` clause)? what is a column alias (within the `select` clause)?
 - Why is a table alias useful? (2 reasons)
 - Why is a column alias useful?
- Computed columns
 - You should be able to read and write queries that project computed columns;
 - Make sure that you understand: whatever computations you choose to project from a `select` statement, projecting those computations does **not** change the contents of the database!
- Aggregate functions

- avg, min, max, sum, count
- Expect to have to read and/or write some of these;
- Where can these be used within a `select` statement?
- What effect do `null` values have with regard to these?
- In a basic `select` statement (with just `from` and `where` clauses), either zero or how many rows will **always** be in the result of a `select` statement projecting an aggregate function?