SQL Review 2/9/2022

**Contents**

[**Entity Relationship Diagrams (ERD)** 3](#_Toc95329116)

[**ERD Minimum Requirements** 3](#_Toc95329117)

[**ERD Creation Thought Process** 3](#_Toc95329118)

[**ERD Reading Thought Process** 4](#_Toc95329119)

[**ERD Exercise** 4](#_Toc95329120)

[**Exercise 1:** 4](#_Toc95329121)

[**Exercise 2:** 4](#_Toc95329122)

[**Exercise 3:** 5](#_Toc95329123)

[**SQL Cheat Sheets and Resources** 5](#_Toc95329124)

[**ERD Tool** 5](#_Toc95329125)

[**SQL Cheat Sheets** 5](#_Toc95329126)

[**Sakila database** 5](#_Toc95329127)

[**Sakila ERD** 5](#_Toc95329128)

[**SQL Exercises** 5](#_Toc95329129)

[**SQL Word Problem Thought Process** 5](#_Toc95329130)

[**Querying Data Tables (SELECT, FROM, SELECT DISTINCT)** 6](#_Toc95329131)

[Exercise 1 6](#_Toc95329132)

[Exercise 2 6](#_Toc95329133)

[Exercise 3 6](#_Toc95329134)

[**Filtering and Organizing Queries (WHERE, LIKE, ORDER BY, ASC, DESC, LIMIT, BETWEEN, MAX, MIN, NULL, AND, OR, NOT, IN, UPPER, LOWER)** 6](#_Toc95329135)

[Exercise 4 6](#_Toc95329136)

[Exercise 5 6](#_Toc95329137)

[Exercise 6 6](#_Toc95329138)

[Exercise 7 6](#_Toc95329139)

[Exercise 8 6](#_Toc95329140)

[Exercise 9 6](#_Toc95329141)

[Exercise 10 6](#_Toc95329142)

[Exercise 11 6](#_Toc95329143)

[Exercise 12 7](#_Toc95329144)

[Exercise 13 7](#_Toc95329145)

[Exercise 14 7](#_Toc95329146)

[Exercise 15 7](#_Toc95329147)

[Exercise 16 7](#_Toc95329148)

[Exercise 17 7](#_Toc95329149)

[Exercise 18 7](#_Toc95329150)

[**Aggregate Functions (COUNT, SUM, GROUP BY, AVG)** 7](#_Toc95329151)

[Exercise 19 7](#_Toc95329152)

[Exercise 20 7](#_Toc95329153)

[Exercise 21 7](#_Toc95329154)

[Exercise 22 7](#_Toc95329155)

[Exercise 23 7](#_Toc95329156)

[Exercise 24 8](#_Toc95329157)

[**Querying Multiple Tables (JOIN, LEFT JOIN, RIGHT JOIN)** 8](#_Toc95329158)

[Exercise 25 8](#_Toc95329159)

[Exercise 26 8](#_Toc95329160)

[Exercise 27 8](#_Toc95329161)

[Exercise 28 8](#_Toc95329162)

[Exercise 29 8](#_Toc95329163)

[Exercise 30 8](#_Toc95329164)

[Exercise 31 8](#_Toc95329165)

[Exercise 32 8](#_Toc95329166)

[Exercise 33 8](#_Toc95329167)

[Exercise 34 8](#_Toc95329168)

[Exercise 35 9](#_Toc95329169)

# **Entity Relationship Diagrams (ERD)**

Every relational database should have an ERD associated with it. When designing a relational database, it is a crucial and necessary step to create an ERD. Not only does the process of creating an ERD help database designers and data engineers understand how to best organize data, but it also helps end-users, such as data scientists/analysts, understand how to quickly and accurately access data that is needed to perform analysis. Any large amount of data needed to perform analysis will most likely need to be stored in a database.

As a data scientist, it will be important to be self-sufficient. Although the responsibilities of how the data arrives to its storage location and/or how it is organized there may not fall under you; you should take ownership of understanding how it is organized, how to access the data and pull the data you need, and also possibly providing feedback to improve data engineering processes that would improve the automation of your analysis.

## **ERD Minimum Requirements**

If you have read this far – heck yeah keep it up.

* Representation of entities (usually tables)
* Representation of relationships between tables (usually lines with either symbols or labels on the lines if symbols are not used)
  + There should be either a symbolic or label-olic description of the type of relationships (one-to-one, one-to-many, many-to-many)
* Primary and foreign keys identified in each entity
  + Usually by either a key symbol or a PK/FK in front of the attribute
* Other attributes labeled in each entity
* A legend to make all symbols/items clear

## **ERD Creation Thought Process**

When creating an ERD, reference the minimum requirements above.

Think about the different objects that your data needs to represent. These objects will most likely represent entities in your ERD. Let’s take any chain of stores as example. Different objects could be employees, the physical stores, the purchases/transactions that occur at the store, and the inventory at the store. Details that *describe* the objects will be the attributes in each table. For example: name, email, and phone number all describe an employee or a customer and could live in those tables. You could also have an email table and a phone table if you wanted to break it out further, but it all depends on how you want to design the database.

Primary keys will need to be created for each table. Usually it is the name of the table followed by id. For example, in the Employee table the Primary Key might be employee\_id.

When thinking about relationships, think to yourself – is it possible for one thing to be related to one or more than one thing from another table. For example, say you have two tables you have identified you need: Store and Employee. Can one store be associated with more than one employee? Or only one employee? Can one employee be associated with more than one store? Or only one? Depends on the business and depends on your data. Should the tables even be connected in the first place? A customer table should not be directly related to an employee table but could be connected through a transaction table.

Once you know the relationships between tables, you have also identified which tables should be directly connected. If two tables are connected, one of them will have a foreign key from the other. Remember: a foreign key is just a primary key that lives in another table. For example, employee\_id is the primary key in the employee table, but it will be the foreign key in the store table. The two tables will then be connected by employee\_id because it will be in both tables.

## **ERD Reading Thought Process**

Reading an ERD should be straightforward if you understand the above.

First thing is first – identify the entities (tables) and the keys within them. The keys will represent how the tables are connected as described above. You will then be able to read how the different tables are related. Look for the symbols or line descriptions to figure out the relationships.

Perform an exercise to try to describe in words how the different tables are related. This will help you understand when you go to query the database, what attributes (columns) you will need to use to join the tables (foreign/primary keys usually).

See if you can trace the lineage from one table to another.

A lot of databases will auto generate an ERD for tables you have created, or they will do the vice versa and will auto-make tables for you if you make the ERD within the tool. Regardless, you might now have timely access to a data engineer or database owner and you might only have the ERD at your fingertips.

# **ERD Exercise**

## **Exercise 1:**

You have been assigned the task of designing a relational database for a used car dealership. Produce an ERD that contains at least 5 entities that have at least 5 attributes each. Follow the guidelines above and make sure your ERD contains the minimum requirements.

## **Exercise 2:**

You have been assigned the task of designing a relational database for a college university. Produce an ERD that contains at least 5 entities that have at least 5 attributes each. Follow the guidelines above and make sure your ERD contains the minimum requirements.

### **Exercise 3:**

You have been assigned the task of designing a relational database for an astronomical lab. Produce an ERD that contains at least 5 entities that have at least 5 attributes each. Follow the guidelines above and make sure your ERD contains the minimum requirements.

# **SQL Cheat Sheets and Resources**

### **ERD Tool**

[ER Diagram Tool | Lucidchart](https://www.lucidchart.com/pages/landing/erd_tool?utm_source=bing&utm_medium=cpc&utm_campaign=_chart_en_us_mixed_search_brand_bmm_&km_CPC_CampaignId=359804038&km_CPC_AdGroupID=1227055131905666&km_CPC_Keyword=%2Blucid%20chart%20%2Berd&km_CPC_MatchType=e&km_CPC_ExtensionID=%7bextensionid%7d&km_CPC_Network=o&km_CPC_AdPosition=&km_CPC_Creative=&km_CPC_TargetID=kwd-76691073618636:loc-190&km_CPC_Country=88859&km_CPC_Device=c&km_CPC_placement=&km_CPC_target=&msclkid=ef5bfc22803f1bde61e39e268069fed9)

* Lucidchart is a tool that can be accessed with a free account
* It can be used to create many different types of diagrams, but we can leverage it to make ERDs

### **SQL Cheat Sheets**

[PowerPoint Presentation (sqltutorial.org)](https://www.sqltutorial.org/wp-content/uploads/2016/04/SQL-cheat-sheet.pdf)​

[SQL-Cheat-Sheet-websitesetup.pdf](https://websitesetup.org/wp-content/uploads/2020/08/SQL-Cheat-Sheet-websitesetup.pdf)

[Ultimate SQL Cheat Sheet 2021 (Download PDF) : Queries, Commands, Etc. | Kanak Infosystems LLP](https://kanakinfosystems.com/blog/sql-cheat-sheet)

### **Sakila database**

<https://demo.phpmyadmin.net/master-config/index.php?route=/database/sql&db=sakila>

### **Sakila ERD**

<https://www.jooq.org/sakila>

# **SQL Exercises**

These exercises have been organized by the different functions you will need to apply to solve them. They can obviously be solved in different ways and maybe even without the function section it falls under; however they are provided to give a hint towards the direction you should be thinking in order to solve it. The questions are organized from easy to harder (but that is subjective to the user).

Paste either your solution below each exercise.

## **SQL Word Problem Thought Process**

When trying to solve a SQL word problem, remember to map out how the tables are connected. First thing you should do is ask yourself “What entities (tables) do I need to solve this problem?” If all the information you need is in one table, then you know you can just write a query for that table. If you need more than one table, then you know you will have to utilize a JOIN statement. Determining which tables you need will depend on what the question is asking for. Figure out which columns you need in your answer and then find the lineage between the tables (if it requires more than one table). Lastly, determine if you need to manipulate the data in any way when returning it – i.e. do I need to perform an aggregate function? Do I need to filter my results? Do I need to order my results?

## **Querying Data Tables (SELECT, FROM, SELECT DISTINCT)**

### Exercise 1

Write a query that will return all of the rows in the Film table. Paste your query below.

### Exercise 2

Write a query that will return only all of the first and last names from the Actor table.

### Exercise 3

Write a query that will return all of the unique postal codes in the sakila database (hint: I am not telling you which table you need to write a query for but use the ERD table to find it out. You only need one table).

## **Filtering and Organizing Queries (WHERE, LIKE, ORDER BY, ASC, DESC, LIMIT, BETWEEN, MAX, MIN, NULL, AND, OR, NOT, IN, UPPER, LOWER)**

### Exercise 4

Write a query that returns all films that an ‘F’ in the title. Retitle the output columns to “f\_titles”.

### Exercise 5

Write a query that returns customer first name and last name ordered by last name descending.

### Exercise 6

Write a query that returns all columns from the Staff table that have an ‘.com’ in the email field. Order the results by first name ascending.

### Exercise 7

Write a query that will return all of the unique postal codes in the sakila database (hint: I am not telling you which table you need to write a query for but use the ERD table to find it out. You only need one table). Order your results by postal code ascending.

### Exercise 8

Write a query that returns the top 10 store IDs in the store table, ordered by store descending.

### Exercise 9

Write a query that returns all payments greater than or equal to $5.00. The result of your query should only include the amount paid and the customer\_id.

### Exercise 10

Write a query that returns all payments greater than or equal to $5.00. The result of your query should only include the customer\_id. Retitle the output columns “greater\_than\_5”

### Exercise 11

Write a query that will return all unique customer IDs with payments between $0 and 10$. Make sure you filter out NULL values in your query.

### Exercise 12

Write a query that will return the max payment from the payment table. Retitle the output column “max\_payment”. Make sure you filter out NULL values in your query.

### Exercise 13

Write a query that will return the earliest customer to have shopped in the database (hint: you will be using the customer table and the create\_date column). Rename the output column to be “Earliest\_Customer\_Date”. Make sure you filter out NULL values in your query.

### Exercise 14

Write a query that returns all columns associated with unique film titles that have AT LEAST one of the following characters in the title: ‘X’ ‘x’ ‘Z’ ‘z’.

### Exercise 15

Write a query that returns all columns associated with unique film titles that MUST HAVE the following characters in the title: ‘A’ ‘F’

### Exercise 16

Write a query that returns only the film title in all lowercase characters.

### Exercise 17

Write a query that returns all first and last names of customers, but make sure that all results are uppercase. Rename the output columns to “upper\_first” and “upper\_last”.

### Exercise 18

Write a query that returns all addresses that DO NOT have a “WA” or “wa” in the name. Order your results by address descending and only display the top 10 results. Also only display results with a “W” or “w” in the name.

## **Aggregate Functions (COUNT, SUM, GROUP BY, AVG)**

### Exercise 19

Write a query that returns the number of rows in the staff table.

### Exercise 20

Write a query that returns the number of rows in the staff table that have a non-null email address.

### Exercise 21

Write a query that returns the total amount of payment in the payment table. Make sure to exclude NULL values.

### Exercise 22

Write a query that returns the average payment of the payment table where the payment is non-null.

### Exercise 23

Write a query that returns total amount of payment by customer ID. Rename the output column “total\_amount”. Order the results by the total\_amount descending.

### Exercise 24

Write a query that returns the total replacement\_cost by films that have an “H” or “h” in the film title. Return only the total\_replacement\_cost.

## **Querying Multiple Tables (JOIN, LEFT JOIN, RIGHT JOIN)**

### Exercise 25

What are the primary and foreign keys of the payment table?

### Exercise 26

If we wanted to connect the staff table to the rental table, what column(s) would we use to JOIN them on?

### Exercise 27

Write a query that returns all staff\_ids and the the rentals they are associated with. Your query result should only include staff\_id, rental\_id, and rental\_date.

### Exercise 28

Write a query that returns all staff\_ids and the the rentals they are associated with. Your query result should only include staff\_id, rental\_id, and rental\_date. Order your results by rental\_date descending.

### Exercise 29

Write a query that returns all staff\_ids and the the rentals they are associated with. Your query result should only include staff\_id, first\_name, last\_name, rental\_id, and rental\_date. Order your results by first\_name ascending. Your results should only include names that have an “E” or “e” in them.

### Exercise 30

If wanted to connect the customer table to the country table, how would you do so? Name the columns in each table you would use.

### Exercise 31

What is the difference between RIGHT JOIN, LEFT JOIN, and JOIN?

### Exercise 32

Write a query that returns all of the customers and the country that they are from. Your result should only output customer\_id and country.

### Exercise 33

Write a query that returns all of the customers and the country that they are from. Your result should only output customer\_id and country. Only include customers associated with a store\_id between 0 and 50.

### Exercise 34

Write a query that returns total number of customers by country that they are from. Your result should only output “total\_customer\_count” and country. Only include customers associated with a store\_id between 0 and 50. Only include customers with an “F” or “f” in the first name. Return the results ordered by total\_customer\_count.

### Exercise 35

Create a question that combines at least 3 tables together in order to get the solution. Explain why it combines 3 tables together and the different keys that are necessary to do so. Swap questions with a classmate and share answers!