
GETTING STARTED SOFTWARE INSTALL - pgAdmin 4

- Google “pgAdmin 4 download windows” and install pgAdmin 4
- Once installed, launch pgAdmin 4 and do the following:
 - Right-click on **Servers** in left pane, select **Create ► Server**
 - On the General tab, name the **Server**: GA Analytics Data
 - On the Connection tab enter **Host address** (for your region):
 - West-coast Region AWS server (includes Austin and Chicago) - analyticsga.cuwj8wuu6wbh.us-west-2.rds.amazonaws.com
 - East-coast Region - analyticsga-east2.c20gkj5cvu3l.us-east-1.rds.amazonaws.com
 - **port**: 5432 **maintenance database**: postgres
 - **username**: analytics_student
 - **password**: analyticsga (check ☒ save password)



WELCOME TO SQL BOOTCAMP

Celia & Craig Fryar

SQL Bootcamp

- ❖ *Introduction to Database & Queries*
- ❖ *Building SELECT statements*
- ❖ *Filtering & Aggregating with WHERE*
- ❖ *Preview Combining Data Tables*

Unit 2: SQL

5: Fundamentals of DB & SQL

6: Filtering & Aggregating

7: Unions and Joins

8: Multiple Joins

9: Data Aggregation

*10: Subqueries Intro***

11: Applying SQL Functions

COURSE INSTRUCTORS



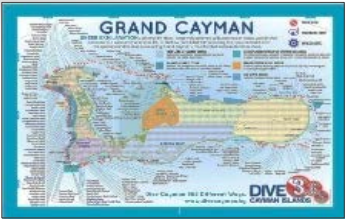
Celia

Coreiger



Data Diver

Coding Geek



BRAIN-TEASER AND CLASS INTRODUCTIONS

ACTIVITY

DIRECTIONS

Work in Groups to solve this puzzle: A farmer with his wolf, his duck and bag of corn come to the east side of a river they wish to cross. There is a boat at the rivers edge, but of course only the farmer can row. The boat can only hold two things (including the rower) at any one time. If the wolf is ever left alone with the duck, the wolf will eat it. Similarly if the duck is ever left alone with the corn, the duck will eat it. How can the farmer get across the river so that all four arrive safely on the other side?



DELIVERABLES

Work together for 5-7 minutes in the Breakout Rooms to solve this brain teaser. Be prepared to **share your answer** when we re-join the main Zoom conference room.

NOTE: if you know the answer, there are two other alternative puzzles to choose from.

BRAIN-TEASER AND CLASS INTRODUCTIONS

Alternative puzzle #2: what do each of these word pictures mean?

DO ¹² OR	CAN'T DO CAN'T DO CAN'T DO CAN'T DO	NO V A T I O N	Ho Ho <u>+Ho</u>	HE AD
WEIGH <u>PRICED</u>	living SHOESTRING	DOUBT	drove IT	1. <i>your blessing</i> 2. <i>your blessing</i> 3. <i>your blessing</i>

ACTIVITY: INTRODUCTIONS & YOUR DATA WORLD



ACTIVITY

DIRECTIONS

1. Let's break into groups and take 10 minutes to...
2. Introduce yourself (name+town+SQL dialect in your data world).
3. Discuss your group's assigned Brain-Teaser.

DELIVERABLES

Let's share, open mic, your group's findings with the Brain-Teasers!



FUNDAMENTALS OF DATABASE AND SQL

Celia Fryar

FUNDAMENTALS OF DATABASES AND SQL

OPENING

FUNDAMENTALS OF DATABASES AND SQL

LEARNING OBJECTIVES

- Market trend toward Self Serve Data Access.
- Review Database structures and the role of SQL.
- Introduce SQL's **SELECT** statement.
- Explore practice data, the Iowa Liquor Sales Database.
- Learn and practice a selection of query command tools, including **DISTINCT, COUNT, AND, OR** and **CAST**.

OF DATABASES AND SQL FUNDAMENTALS

INTRODUCTION

BASIC SQL

- SQL ('si:kwəl) , Structured Query Language. Developed at IBM by Donald Chamberlin (IDS) and Raymond Boyce after learning about the new relational model from “Ted” Codd (Boyce-Codd normal form) in early '70s.
- There are a number of implementations of SQL databases implementations, including:
 - PostgreSQL (what we will be using)
 - MySQL
 - MS SQL
 - Oracle
- Each has its own proprietary additions to a common core of commands and structures, based on ANSI standards.



EXCEL vs SQL

Why do we need SQL when we have Excel?

- Excel is limited by available memory and system resources.
- Excel also has a **limit** of **1,048,576 rows** and **16,384 columns**.

In other words, Excel is a local tool that cannot manage or interact with very *large* datasets. *This is when SQL steps in!*

BASIC SQL

What can SQL do that Excel can't...

- SQL can rapidly navigate databases, and can query, retrieve and aggregate millions of records.
- SQL is also more adept at creating data flows for cleaning and preparing data at high volumes than Excel.
- SQL is *the* industry standard for data query and retrieval.

Excel and SQL both use many of the similar structures and functions (COUNT, IF THEN ELSE, SUM, etc.).

BASIC SQL: PostgreSQL

We will be using PostgreSQL for this class because:

- It's object oriented, relational database management system (DBMS);
- It's powerful and standardized;
- It's free and open-source;

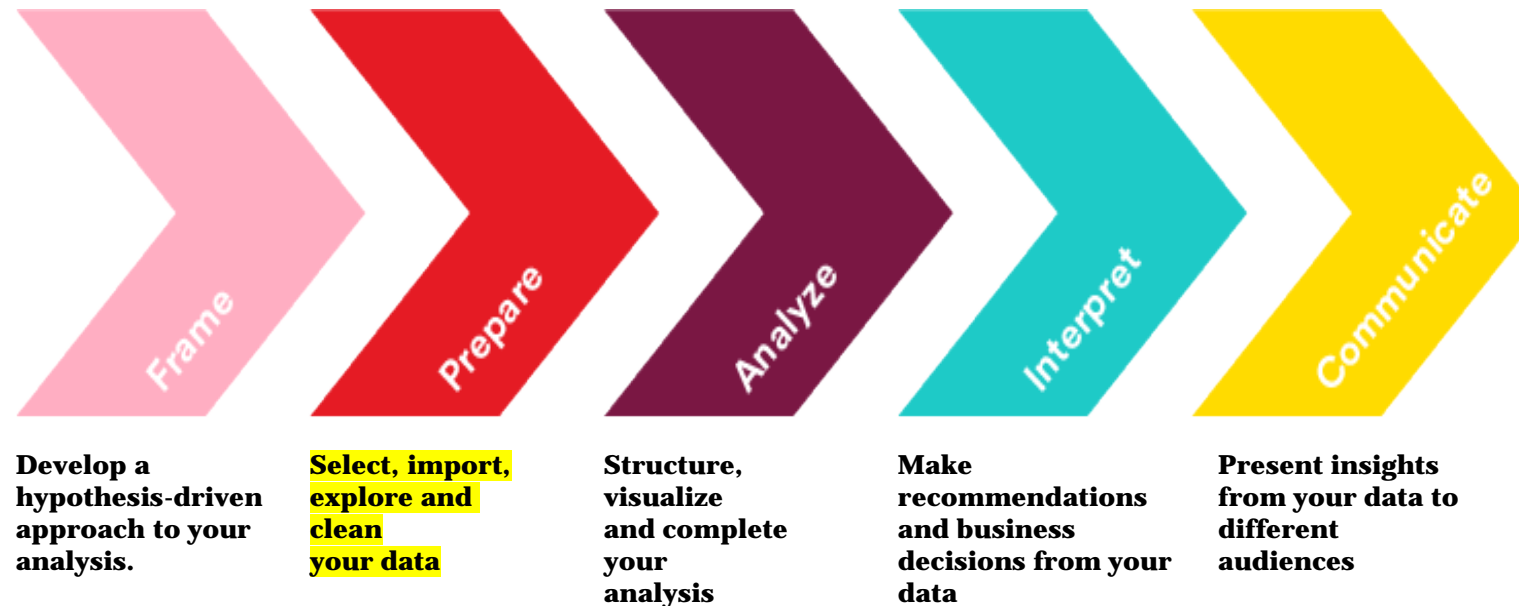
The skills learned here are applicable to other implementations.

- What version (or vendor) of SQL exist in your world?
- Do you work (or have you worked) with relational tables/dbms?



BASIC SQL: ANALYTICS WORKFLOW

Obtaining data often means accessing a **SQL server** to retrieve the data you need. We'll use SQL in the second step of the GA data workflow — preparing your data: **selecting, exploring, and cleaning**.



FUNDAMENTALS OF DATABASES AND SQL

PRACTICE DATASET: IOWA LIQUOR SALES DATABASE

Unsaved View

Save As...

Revert

Based on Iowa Liquor Sales
This dataset contains the spirits purchase information of Iowa Class "E" liquor licensees by product and date of purchase from January 1, 2012 to current. The dataset can be used to analyze

Manage

More Views

Filter

Visualize

Export

Discuss

Embed

About

Find in this Dataset

	Invoice/Item Number	Date	Store Number	Store Name	Address	City	Zip Code	Store Location	County Number	County
1	INV-04039700001	03/31/2017	5378	Casey's General Store # 2543	201 S Main St.	Albion	50005	(42.109312°, -92.9564°)	64	MARSHALL
2	INV-04111400036	03/31/2017	4073	Uptown Liquor, LLC	306 Hwy 69 South	Forest City	50436	(43.261538°, -93.6395°)	95	WINNEBAGO
3	INV-04021400001	03/31/2017	3728	Brewski's Beverage	726 Creek Top	Council Bluffs	51503	(41.262316°, -95.8578°)	78	POTTAWAMITTIE
4	INV-04100700038	03/31/2017	4823	Casey's General Store #3223	903 N Sumner Ave	Creston	50801	(41.066824°, -94.3788°)	88	UNION
5	INV-04100600043	03/31/2017	2553	Hy-Vee Food Store / Creston	600 Sheldon	Creston	50801	(41.051313°, -94.3788°)	88	UNION
6	INV-04100600011	03/31/2017	2553	Hy-Vee Food Store / Creston	600 Sheldon	Creston	50801	(41.051313°, -94.3788°)	88	UNION
7	INV-04100800026	03/31/2017	4291	Fareway Stores #597 / Creston	105 E Adams St	Creston	50801	(40.71572°, -94.2388°)	88	UNION
8	INV-04100800017	03/31/2017	4291	Fareway Stores #597 / Creston	105 E Adams St	Creston	50801	(40.71572°, -94.2388°)	88	UNION
9	INV-04111000040	03/31/2017	2585	Hy-Vee Wine and Spirits / Algona	1516 Highway 169 North	Algona	50511	(43.081288°, -94.2355°)	55	KOSSUTH
10	INV-04111000056	03/31/2017	2585	Hy-Vee Wine and Spirits / Algona	1516 Highway 169 North	Algona	50511	(43.081288°, -94.2355°)	55	KOSSUTH
11	INV-04110700007	03/31/2017	3682	Jumbo's	110 E Main St	Wesley	50483	(43.088185°, -93.9855°)	55	KOSSUTH
12	INV-04111000009	03/31/2017	2585	Hy-Vee Wine and Spirits / Algona	1516 Highway 169 North	Algona	50511	(43.081288°, -94.2355°)	55	KOSSUTH
13	INV-04110300013	03/31/2017	5145	South Side Food Mart	1101 Army Post Rd	Des Moines	50315	(41.526511°, -93.6277°)	77	POLK
14	INV-04110100012	03/31/2017	5131	Oasis	1401, Buchanan St	Des Moines	50316	(41.598791°, -93.5577°)	77	POLK
15	INV-04110000017	03/31/2017	3698	Ingersoll Wine Merchants	1300 50th St	West Des Moines	50266	(41.590889°, -93.7777°)	77	POLK
16	INV-04110000009	03/31/2017	3698	Ingersoll Wine Merchants	1300 50th St	West Des Moines	50266	(41.590889°, -93.7777°)	77	POLK
17	INV-04109900001	03/31/2017	4669	Vom Fass / Des Moines	833, 42nd St	Des Moines	50312	(41.593431°, -93.6777°)	77	POLK
18	INV-04098800023	03/31/2017	2607	Hy-Vee Wine and Spirits / Shenandoah	520 So Fremont	Shenandoah	51601	(40.761736°, -95.3873°)	73	PAGE
19	INV-04109600037	03/31/2017	4597	Quik Trip #523 / Army Post Dr	850 Army Post Rd	Des Moines	50315	(41.526428°, -93.6277°)	77	POLK
20	INV-04109600029	03/31/2017	4597	Quik Trip #523 / Army Post Dr	850 Army Post Rd	Des Moines	50315	(41.526428°, -93.6277°)	77	POLK
21	INV-04109500020	03/31/2017	4478	U S Gas	3000, SW 9th St	Des Moines	50315	(41.557862°, -93.6277°)	77	POLK
22	INV-04109500005	03/31/2017	4478	U S Gas	3000, SW 9th St	Des Moines	50315	(41.557862°, -93.6277°)	77	POLK
23	INV-04109300037	03/31/2017	5245	Phillips 66	1516, SE 1st St	Des Moines	50315	(41.575862°, -93.6177°)	77	POLK
24	INV-04109300029	03/31/2017	5245	Phillips 66	1516, SE 1st St	Des Moines	50315	(41.575862°, -93.6177°)	77	POLK
25	INV-04109100066	03/31/2017	3443	Super Saver Iv	1141 N Broadway	Council Bluffs	51503	(41.270881°, -95.8378°)	78	POTTAWAMITTIE
26	INV-04109100059	03/31/2017	3443	Super Saver Iv	1141 N Broadway	Council Bluffs	51503	(41.270881°, -95.8378°)	78	POTTAWAMITTIE

Visualize

Calendar

Map

Chart

Visualization Type Tree Map

Column

Stacked Column

Bar

Stacked Bar

Pie

Donut

Line

Area

Timeline

Bubble

Tree Map

Data Selection

Configuration

Choose label data

No column selected

DogsCats

Choose value data

No column selected

12%25%

Data Presentation

Colors

Chart Details

FUNDAMENTALS OF DATABASES AND SQL

BASIC SQL: SELECT STATEMENT

BASIC SQL: SELECT STATEMENT PREVIEW

- ✓ **SELECT** the columns
- ✓ **FROM** *points* to the table
- ✓ **WHERE** *filters* on rows
- ✓ **GROUP BY** *aggregates* across values of a variable
- ✓ **HAVING** *filters* groups
- ✓ **ORDER BY** *sorts or arranges* the results
- ✓ **LIMIT** *limits* result to the first n rows

BASIC SQL: QUESTIONS TO CODE

Example question #1:

Tell me which **products** in our inventory are varieties of **Scotch Whiskies**.

Include **all of the details** we have for each item.

And if you don't mind, put them into **alphabetical order**, please.

```
SELECT *  
FROM products  
WHERE category_name = 'SCOTCH WHISKIES'  
ORDER BY item_description ASC;
```

BASIC SQL: QUESTIONS TO CODE

Example question #2:

Show a **tally** of **active** and **inactive storefronts** in the Iowa sales territory.

Display the list arranged by **store status**.

```
SELECT store_status,  
       COUNT(STORE) AS store_headcount  
FROM stores  
GROUP BY store_status;
```

BASIC SQL: QUESTIONS TO CODE

Example question #2:

Show a **tally** of **active** and **inactive storefronts** in the Iowa sales territory.
Display the list arranged by **store status**.

Data Output		Explain	Messages	Query History
	store_status text	store_headcount bigint		
1	I	548		
2	A	1425		

BASIC SQL: QUESTIONS TO CODE

Example question #3:

The Sales Manager wants to know top performing **stores** by **revenue**.

List the top 10 locations by **store number**, total **sales** and **transactions**.

```
SELECT store,  
       SUM(total) AS total_sales,  
       COUNT(total) AS total_transactions  
FROM sales  
GROUP BY store  
ORDER BY total_sales DESC  
LIMIT 10;
```

BASIC SQL: QUESTIONS TO CODE

Example question #3:

The Sales Manager wants to know top performing **stores** by **revenue**.

List the top 10 locations by **store number**, total **sales** and **transactions**.

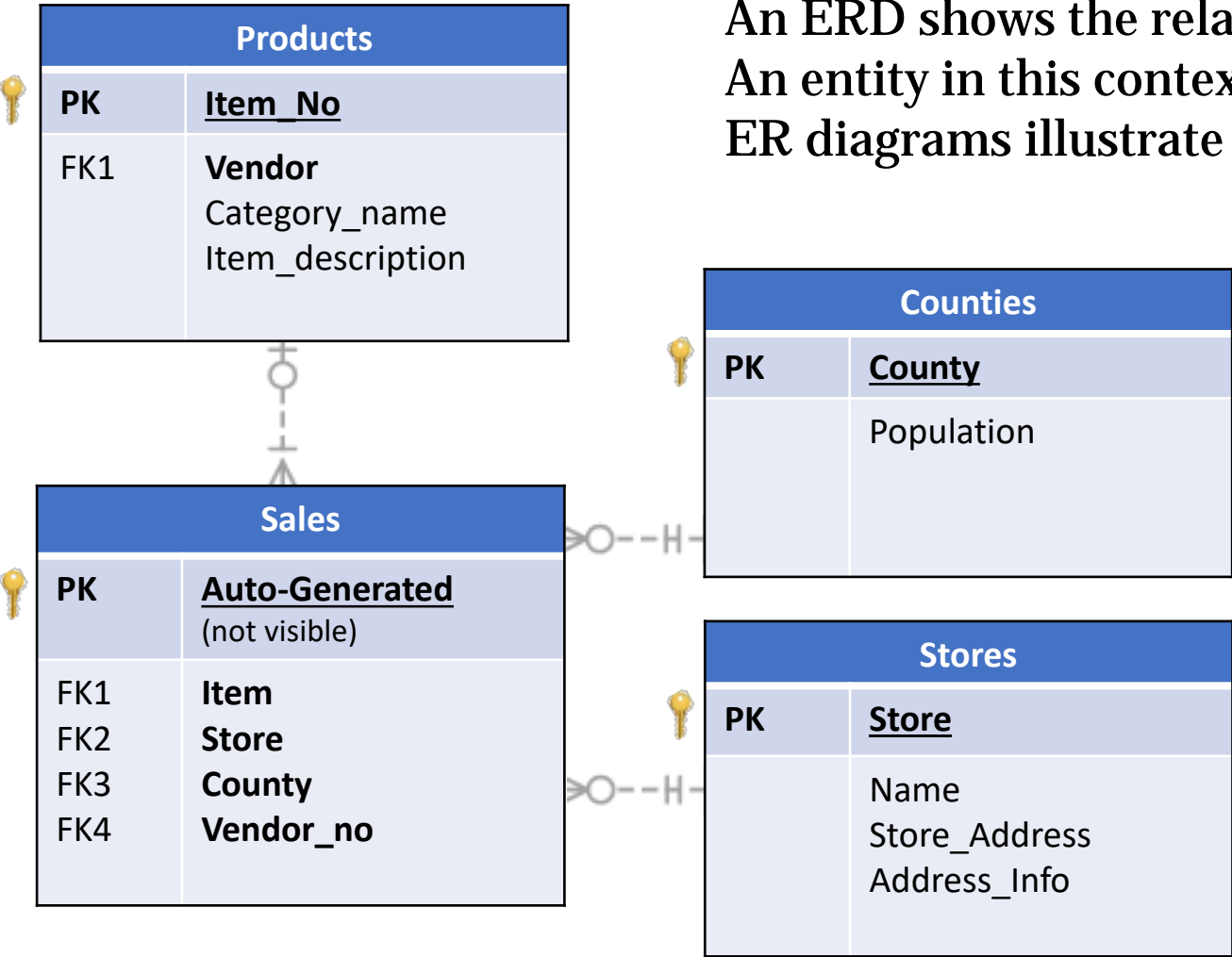
	Data Output	Explain	Messages	Query History
	store integer	total_sales numeric	total_transactions bigint	
1	2633	13920087.22	29465	
2	4829	11942399.97	25827	
3	3420	6159480.06	5568	
4	3385	5734721.57	7135	
5	2512	5665143.70	24038	
6	3814	4907465.88	2688	
7	3952	4289169.59	13415	
8	3354	3308625.56	5306	
9	2625	3169984.14	13843	
10	3773	3129506.57	12608	

FUNDAMENTALS OF DATABASES AND SQL

DATA DESCRIPTION

Download and Open: Data Description - Iowa Liquor Sales DB.pdf

DATA DESCRIPTION: ENTITY RELATIONSHIP DIAGRAM (ERD)



An ERD shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

PK: Primary Key
FK: Foreign Key

NAVIGATION GUIDED PRACTICE

- If we want to view the contents of a table, we should:
 - Navigate to the tables of the public schema of the Iowa database.
 - There are four tables: **products, sales, stores, counties**
 - To view data, right click on the table, go to View/Edit Data, and select Top 100 rows.

NAVIGATION GUIDED PRACTICE: Data Preview

- Open the connection to the **server** name, by clicking on + mark.
- Here is the directory tree that we will need to access the **data tables**:
 - Servers (*pgAdmin can be configured for more than one server config*)
 - Databases
 - Iowa_Liquor_Database (*our database for today*)
 - Schemas
 - Public
 - Tables
 - Columns

We will look at the **Object** browser tab and navigate it to explore how the database's contents and organization.

FUNDAMENTALS OF DATABASES AND SQL

INDEPENDENT PRACTICE: IOWA LIQUOR DATABASE

UNDERSTANDING THE IOWA LIQUOR SALES DATABASE



ACTIVITY

DIRECTIONS

1. Look through all Iowa tables and explore the data in those tables.
2. Take 10 mins and make notes about the database answering the “deliverables” questions below.

DELIVERABLES

1. Characterize each table as either **transactional** or **reference**.
2. Write a few sentences describing the data stored in each table.
Note the **data types** assigned to each column.
3. What **columns** do you see that could serve as links between tables later in our data exploration?

FUNDAMENTALS OF DATABASES AND SQL

GUIDED PRACTICE: BASIC SQL

BASIC SQL

Common best practices to keep in mind.

COMMON PUNCTUATION

- Signal the end of your SQL Query with a semicolon (;)
- Commas separate column names in output list (,)
- Use single quotations around text criteria ('Tequila').

QUERY CODE SPACING

- Layout your query for readability, collaboration and debugging.
- SQL only requires a single white space to separate elements.

NOTES WITHIN QUERIES

- Always provide comments (source, revision, author, etc.)
- Double dash (--) comments to end of line
- Multiple lines provided by the pair /* */

BASIC SQL

SQL is used to communicate questions to the database. The three main clauses are **SELECT**, **FROM** and **WHERE**.

SELECT

- Allows you to select certain columns from a table.
- Determines which columns of information are in output.

FROM

- Specifies the tables from which the query extracts data.
- Defines the relationships to combine tables (join conditions).

WHERE

- Applies filters and criteria to the data in the tables.

BASIC SQL

- When we selected the top 100 rows of a table, like products, using the View Data menu or Query Tool menu selections, we actually ran a SQL statement in the background:

```
SELECT * FROM products LIMIT 100;
```

- Let's think through these questions:
 - What does * mean?
 - What does **FROM** products mean?
 - What does the **LIMIT** do?

BASIC SQL: SELECT

For the remainder of the lesson, we will be writing queries in the pgAdmin Query Tool window (*Tools dropdown menu*).

- We may query all columns with **SELECT *** or query for specific ones.
- Columns are presented in the order of the **SELECT** query line.
- We can tell **SELECT** which columns or variables we want:

SELECT * FROM products **LIMIT 100;**

SELECT item_no, item_description **FROM** products;

SELECT store, store_address **FROM** stores;

BASIC SQL: ORDER BY

- **ORDER BY** sorts results in an ascending or descending fashion.
- Specify the column by name or by sequence number.
- The default sort order is ascending, and you may also specify ascending (**ASC**) or descending (**DESC**) to determine the sort order.

```
SELECT * FROM products WHERE case_cost >= 100 ORDER BY 1;
```

- Change the sort to confirm boundaries – or to change presentation priority.

BASIC SQL: FILTER WITH WHERE

- The **WHERE** clause filters rows by the criteria it contains.
- Numeric comparisons: >, >=, <, <=, =, != and <>
- String comparison: **ILIKE** and **LIKE (NOT LIKE)**

```
SELECT * FROM products  
    WHERE category_name ILIKE 'SCOTCH WHISKIES';
```

```
SELECT * FROM products WHERE case_cost >= 100;
```

```
SELECT vendor_name FROM products  
    WHERE category_name NOT ILIKE 'SCOTCH WHISKIES';
```

```
SELECT * FROM products WHERE shelf_price != 50;
```

BASIC SQL: COMPOUNDED WHERE

When the desired filter criteria have more than one factors, create compound **WHERE** clauses by using the logical operators:

- **AND**: Returns TRUE if both conditions are true.
- **OR**: Returns FALSE if neither condition is true (TRUE if either is true).

Example:

```
SELECT vendor_name FROM products
WHERE category_name ILIKE 'SCOTCH WHISKIES'
AND (case_cost < 50 OR case_cost > 100);
```

BASIC SQL: DISTINCT

- **DISTINCT** further defines the results by eliminating exact duplicates. as a data cleaning tool – or an aggregator!

```
SELECT DISTINCT *  
FROM products;
```

- We can add **DISTINCT** to the query statement to eliminate duplicates:

```
SELECT DISTINCT category_name, vendor_name  
FROM products;
```

- Adding **ON** focuses the action of **DISTINCT** onto one column:

```
SELECT DISTINCT ON (category_name) category_name, count(vendor_name)  
FROM products  
GROUP BY category_name;
```

BASIC SQL: DATA VALIDATION

▸ COUNT

- Returns the number of rows that matches some specified criteria.
- If criteria includes only a column name, returns the number of non-NULL values in that column.
- Syntax: **COUNT**(field1)

▸ Example:

```
SELECT category_name, COUNT(item_no)
FROM products
GROUP BY category_name
LIMIT 100;
```

More on this later....

BASIC SQL: CAST()

Some information is stored in a form that creates errors when queries compare different data types, because you can only evaluate data of same types.

Syntax: **CAST** (column_name **AS** new data type)

Converts the contents of “field” to the specified data type to enable comparisons. May be aliased to rename column output

- Question: In the Products table, what data type is proof?
- **CAST(proof AS numeric)**

FUNDAMENTALS OF DATABASES AND SQL

INDEPENDENT PRACTICE: BASIC SQL

INDEPENDENT PRACTICE: BASIC SQL



EXERCISE

DIRECTIONS

Let's answer the following questions by writing and executing SQL queries:

1. Which products come in packs larger than 12? How many unique products have less than 12 in a pack?
2. Which products have a case price of less than \$70?
3. Which products come in packs larger than 12 AND have a case_cost of less than \$70?
4. Which types of products have a proof of 85 or more?
5. Which products are scotch whiskies OR are over 85 proof?
6. How many stores are active (use store_status)? Inactive?

FUNDAMENTALS OF DATABASES AND SQL

SYNTAX RECAP

- **SELECT** designates output; * for all columns or specific column names separated by commas; may span multiple lines for readability and clarity.
- **FROM** table(s) to be used for data source
- **WHERE** opportunity to filter raw data leveraging AWS servers; each clause must contain the column name and the filtering criteria; may compound criteria with **AND**, **OR** and other SQL tools.
- **DISTINCT** follows **SELECT** to either eliminate exact duplicate records for data cleaning, or to narrow the focus for unique combinations; **DISTINCT ON(column_name)** limits the focus of uniqueness to one column.
- **COUNT(column_name)** creates an aggregated numeric count or counts all records if count (*)
- **CAST(column_name AS data_type)** converts the contents of “field” to the specified data type to enable comparisons for that instance. Must evaluate data of same types.
Example: **CAST(proof AS numeric)** for a number stored as text.
- **ORDER BY** and **LIMIT** are last in the order of commands.

FUNDAMENTALS OF DATABASES AND SQL

CONCLUSION

FUNDAMENTALS OF DATABASES AND SQL

RECAP

- Discussed various database structures.
- Introduced SQL – PostgreSQL – pgAdmin 4.
- Connected to an online AWS database for this unit.
- Described the data's contents and connections (*story of one row*).
- Learned how to run **SELECT** statements with **WHERE** clauses.
- Introduced **DISTINCT**, **COUNT**, **AND**, **OR** and **CAST**
- Explored SQL grammar, syntax and punctuation.

FUNDAMENTALS OF DATABASES AND SQL

Q&A

FUNDAMENTALS OF DATABASES AND SQL

RESOURCES

FUNDAMENTALS OF DATABASES AND SQL

RESOURCES

- AND OR:

https://www.techonthenet.com/sql/and_or.php

- CAST Function:

<https://msdn.microsoft.com/en-us/library/ms187928.aspx>

- The evolution of SQL: Chamberlin, Donald (2012). ["Early History of SQL"](#). *IEEE Annals of the History of Computing*. **34** (4): 78–82.
Retrieved 3 February 2018.