<u>Intro to SQL and Databases Bootcamp</u>

Student Handout

Instructor: Michael Valeri

Handout 1: valerianalytics database schema

ALBUMS TABLE SCHEMA

field/column	data type	sample data
row	integer	1
artist	text	Red Hot Chili Peppers
album	text	Californication
release_date	date	6/8/1999
genre	text	Rock
plays	integer	120
rating	numeric(10,2)	4
org_price	money	$11.99
market_value	numeric(10,2)	3.2
burned	boolean	FALSE
playable	boolean	TRUE

NORTH_KOREAN_MISSLE_TESTS TABLE SCHEMA

field/column	data type	sample data
row_id	integer	5
date_launched	date	9/1/1984
date_entered	date	12/23/2016
launch_time_utc	date	(null)
missle_name	varchar	Scud-B
missle_type	varchar	SRBM
launch_authority	varchar	(null)
facility_name	varchar	Tonghae Satellite
facility_location	varchar	Hwadae County
facility_other_name	varchar	Musudan-ri
facility_latitude	float	40.8499966
facility_longitude	float	129.666664
landing_location	varchar	(null)
apogee	varchar	200 km
distance_traveled	varchar	(null)
confirmation_status	varchar	Confirmed
test_outcome	varchar	Failure
additional_info	varchar	(null)

PRODUCTS TABLE SCHEMA

field/column	data type	sample data
item_no	integer	904616
category_name	text	TEQUILA
item_description	text	Jose Cuervo
vendor	integer	305
vendor_name	text	Mhw Ltd
bottle_size	integer	750
pack	integer	12
inner_pack	integer	1
age	text	(null)
proof	text	40
list_date	timestamp	2/11/2009
bottle_price	money	$9.77
shelf_price	numeric(10,2)	14.66
case_cost	numeric(10,2)	117.22

COUNTIES TABLE SCHEMA

field/column	data type	sample data
county	text	Adair
population	integer	7682

STORES TABLE SCHEMA

field/column	data type	sample data
store	integer	2106
name	text	Hillstreet News and Tobacco
store_status	text	A
store_address	text	2217 CollegeCedar Falls, IA
address_info	text	(null)

SALES TABLE SCHEMA

field/column	data type	sample data
date	timestamp	6/27/2014
convenience_store	text	Y
store	integer	4771
county_number	county_number	57
county	text	Linn
category	text	1081600
category_name	text	WHISKEY LIQUEUR
vendor_no	text	421
vendor	text	Sazerac Co. Inc.
item	integer	64858
description	text	Fireball Cinnamon Whiskey
pack	integer	1
liter_size	integer	3000
state_btl_cost	money	29.72
btl_price	money	44.58
bottle_qty	integer	1
total	numeric(10,2)	44.58

Draw the ERD (Entity Relationship Diagram) between all the tables in 'valerianalytics' db

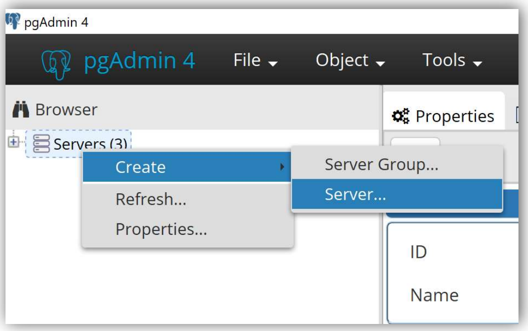
Handout 2: Connecting to 'valerianalytics' database

Step 1 - Download most recent version of PG-Admin 4 and open (Mac OS Users - If getting error "PgAdmin4 is damaged and cannot be opened, click troubleshoot link"

Windows	https://www.pgadmin.org/download/pgadmin-4-windows/
Mac*	https://www.pgadmin.org/download/pgadmin-4-macos/
MAC Troubleshoot	http://www.tech-recipes.com/rx/45404/mac-downloaded-app-is-damaged-and-cant-be-opened-error-solved/

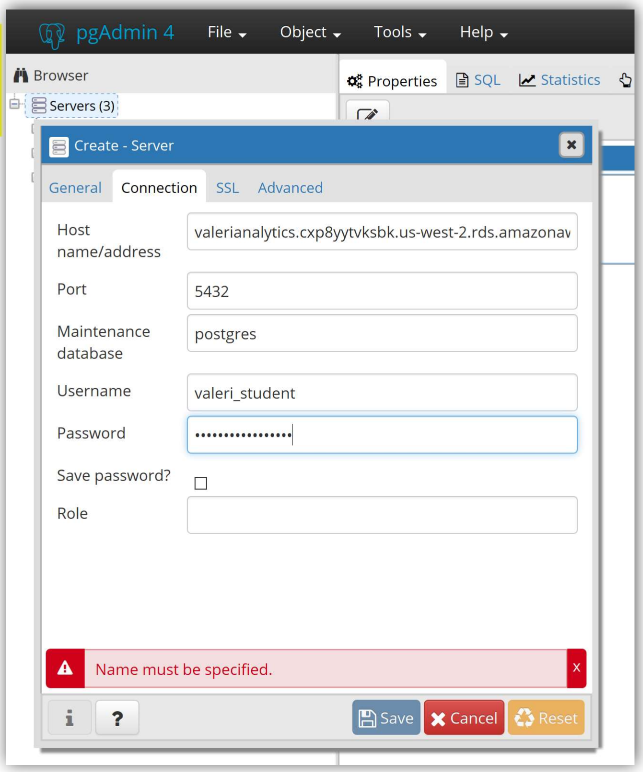
Step 2 - Get connected to 'valerianalytics' db (Copy and paste)

A. Right click "Servers" and then select "Create --> Server..."



B. Navigate to the "Connection" tab, then enter in the following credentials

Hostname/address	valerianalytics.cxp8yytvksbk.us-west-2.rds.amazonaws.com
Port	5432
Maintenance_DB	postgres
Username	valeri_student
Password	analyticsvaleri



Host name/address valerianalytics.cxp8yytvksbk.us-west-2.rds.amazonav

Port 5432

Maintenance database postgres

Username valeri_student

Password ••••••••••••••••

Save password? □

Role

Name must be specified.

Handout 3: Basic Syntax of SQL

	Definition	Example of what goes here
1 **SELECT**		
2 **FROM**		
3 **WHERE**		
4 **GROUP BY**		
5 **HAVING**		
6 **ORDER BY**		
7 **LIMIT**		

Key Takeaways: Basic Syntax of SQL

Handout 4: Filtering with the WHERE Clause

				WHAT COLUMN/FIELD DATA TYPE ARE YOU FILTERING ON?				
USE CASE	OPERATOR	DESCRIPTION	WHEN SHOULD I USE	*BOOLEAN EXAMPLE*	*CHARACTER EXAMPLE*	*NUMERIC EXAMPLE*	*DATE EXAMPLE*	*DATE WITH TO_CHAR EXAMPLE*
1	=	Equal (single criteria)	**You want to filter on 1 column on 1 condition**	PLAYABLE = TRUE	ALBUM = 'Californication'	ORG_PRICE = 9.99	RELEASE_DATE = '2004-02-10'	TO_CHAR(RELEASE_DATE, 'YYYY') = '2009'
	>	Greater than (single criteria)		▨	▨	ORG_PRICE > 9.99	RELEASE_DATE > '2004-02-10'	TO_CHAR(RELEASE_DATE, 'YYYY') > '2009'
	<	Less than (single criteria)		▨	▨	ORG_PRICE < 9.99	RELEASE_DATE < '2004-02-10'	TO_CHAR(RELEASE_DATE, 'YYYY') < '2009'
	>=	Greater than or equal (single criteria)		▨	▨	ORG_PRICE >= 9.99	RELEASE_DATE >= '2004-02-10'	TO_CHAR(RELEASE_DATE, 'YYYY') >= '2009'
	<=	Less than or equal (single criteria)		▨	▨	ORG_PRICE <= 9.99	RELEASE_DATE <= '2004-02-10'	TO_CHAR(RELEASE_DATE, 'YYYY') <= '2009'
	<> or !=	Not equal		PLAYABLE != FALSE	ALBUM != 'Californication'	ORG_PRICE != 9.99	RELEASE_DATE != '2004-02-10'	TO_CHAR(RELEASE_DATE, 'YYYY') != '2009'
	LIKE	Look for a specified pattern in a column		▨	ARTIST LIKE '%K'	▨	▨	▨
	NOT LIKE	Look for a specified pattern in a column (Not Like)		▨	ARTIST NOT LIKE '%K'	▨	▨	▨
2	BETWEEN	Between two numeric values or dates (multiple criteria)	**You want to filter between two ranges (numeric or date)**	▨	▨	ORG_PRICE BETWEEN 9.99 AND 12.99	RELEASE_DATE BETWEEN '2001-01-01' AND '2009-01-01'	TO_CHAR(RELEASE_DATE, 'YYYY') BETWEEN '2009' and '2011'
3	AND	Logical operator AND	**You want to filter on multiple columns**	PLAYABLE = TRUE AND GENRE = 'Rap'	ARTIST = 'Kanye West' AND PLAYABLE = TRUE	9.99	RELEASE_DATE >= '2001-01-01' AND RELEASE_DATE <= '2009-01-01'	(same as left)
	OR	Logical operator OR		PLAYABLE = TRUE OR GENRE = 'Metal'	ARTIST = 'Kanye West' OR RATING = 5	9.99	RELEASE_DATE >= '2001-01-01' OR GENRE = 'Electronic'	(same as left)
4	IS	Logical operator for (null) records	**You are looking for null values in a column**	PLAYABLE IS NULL	ARTIST IS NULL	ORG_PRICE IS NULL	RELEASE_DATE IS NULL	(same as left)
	IS NOT	Logical operator for not (null) records		PLAYABLE IS NOT NULL	ARTIST IS NOT NULL	ORG_PRICE IS NOT NULL	RELEASE_DATE IS NOT NULL	(same as left)
5	IN	Equal (multiple criteria)	**You want to filter on 1 column with multiple conditions**	BURNED IN (TRUE, FALSE)	ALBUM IN ('Californication', 'By the Way')	ORG_PRICE IN (9.99, 12.99)	RELEASE_DATE IN ('2012-01-01', '2017-01-02')	TO_CHAR(RELEASE_DATE, 'YYYY') IN ('2009' ,'2011')
	NOT IN	Not equal (multiple criteria)		BURNED NOT IN (TRUE, FALSE)	ALBUM NOT IN ('Californication', 'By the Way')	ORG_PRICE NOT IN (9.99, 12.99)	RELEASE_DATE NOT IN ('2012-01-01', '2017-01-02')	TO_CHAR(RELEASE_DATE, 'YYYY') NOT IN ('2009' ,'2011')

Key Takeaways: Filtering with the WHERE Clause

Handout 5: Aggregations and GROUP BY and Fix the Code

Key Takeaways: Aggregations and Group BY

Key Takeaways: Fix the Code

Handout 6a: Joining Tables in SQL

1. Why do we join tables in SQL?

2. What are the 3 requirements to join tables in SQL?

3. What are the 3 rules of the JOIN syntax?

4. How do we visualize a JOIN?

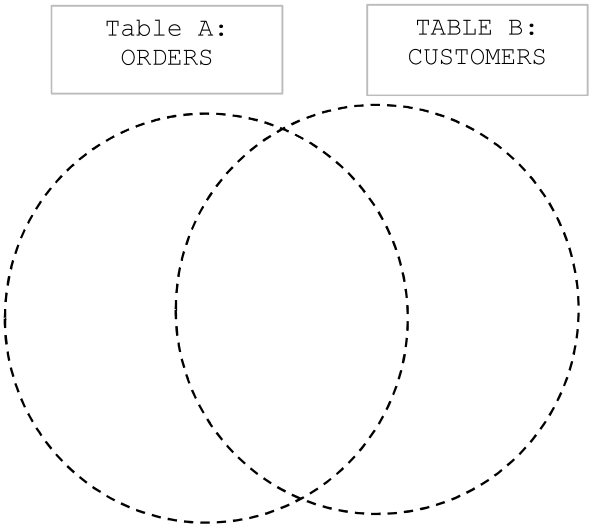
5. How do you join two tables that don't have the same column in common?

5. What do you do if you are getting duplicate records?

Handout 6b: Which JOIN do I use? (Class Exercise)

STEP 1

Make a joined table of order_id, cust_id, and state. Put CUST_IDs in correspondig circle



	ORDER_ID	CUST_ID	STATE
1			
2			
3			
4			
5			
6			
7			

STEP 2

Make a table of the count of orders and customers in each state. Don't count NULL customer_id or order_id

ORDER COUNT	CUSTOMER COUNT	STATE

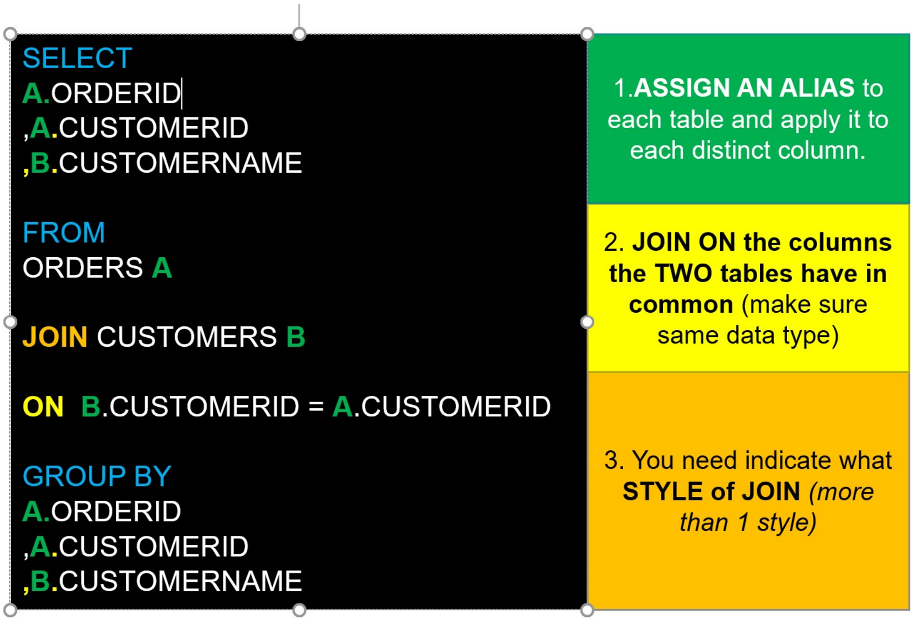
STEP 3

Tell us the total count of orders and customers overall.

ORDER COUNT	CUSTOMER COUNT

Handout 6c: Supplemental Join Notes

JOIN Syntax for two tables



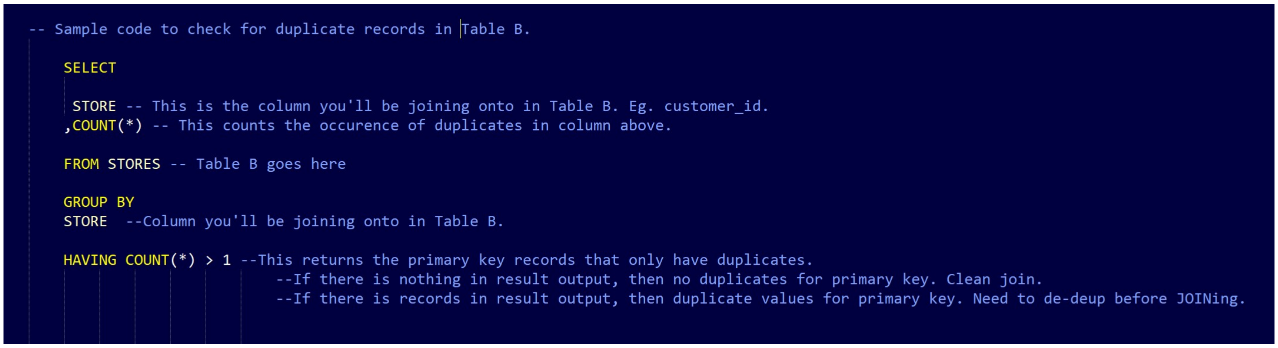
```
SELECT
A.ORDERID
,A.CUSTOMERID
,B.CUSTOMERNAME

FROM
ORDERS A

JOIN CUSTOMERS B

ON  B.CUSTOMERID = A.CUSTOMERID

GROUP BY
A.ORDERID
,A.CUSTOMERID
,B.CUSTOMERNAME
```

1.**ASSIGN AN ALIAS** to each table and apply it to each distinct column.

2. **JOIN ON the columns the TWO tables have in common** (make sure same data type)

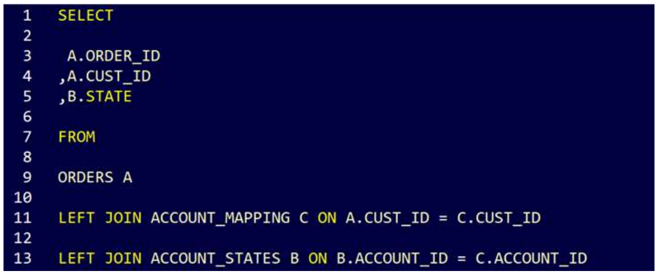
3. You need indicate what **STYLE of JOIN** *(more than 1 style)*

Sample Code for Checking for Duplicate Records in Table B



```
-- Sample code to check for duplicate records in Table B.

    SELECT

     STORE -- This is the column you'll be joining onto in Table B. Eg. customer_id.
    ,COUNT(*) -- This counts the occurence of duplicates in column above.

    FROM STORES -- Table B goes here

    GROUP BY
    STORE  --Column you'll be joining onto in Table B.

    HAVING COUNT(*) > 1 --This returns the primary key records that only have duplicates.
                            --If there is nothing in result output, then no duplicates for primary key. Clean join.
                            --If there is records in result output, then duplicate values for primary key. Need to de-deup before JOINing.
```

Sample Code for Daisy Chaining (For joining on intermediary table)



```
SELECT

 A.ORDER_ID
,A.CUST_ID
,B.STATE

FROM

ORDERS A

LEFT JOIN ACCOUNT_MAPPING C ON A.CUST_ID = C.CUST_ID

LEFT JOIN ACCOUNT_STATES B ON B.ACCOUNT_ID = C.ACCOUNT_ID
```