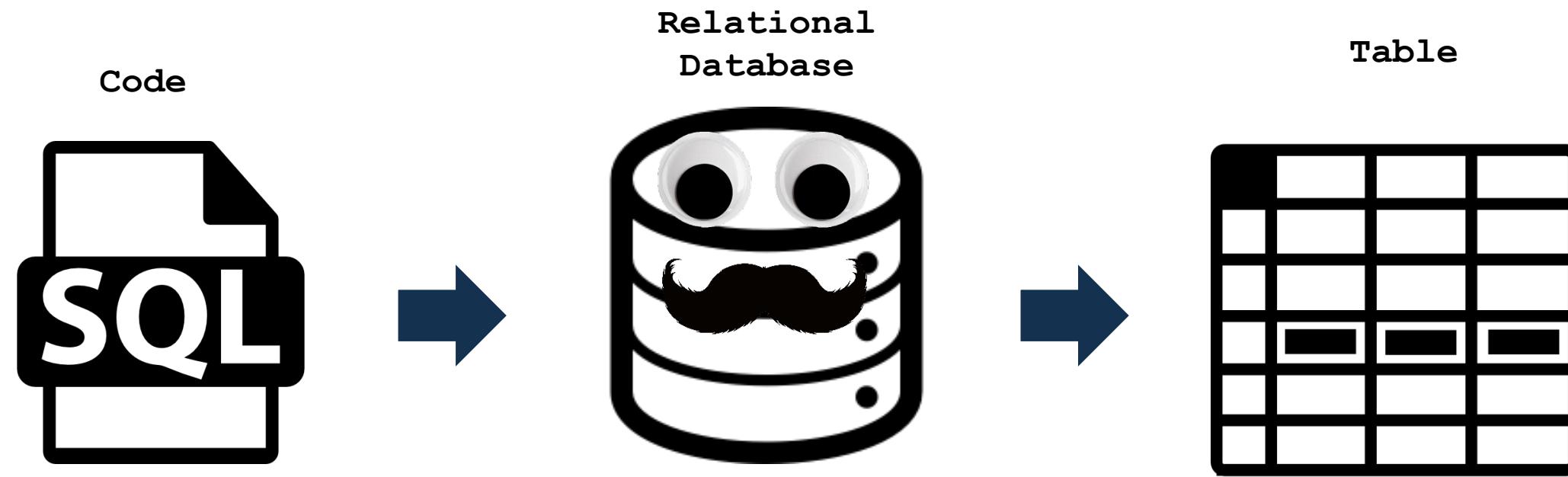




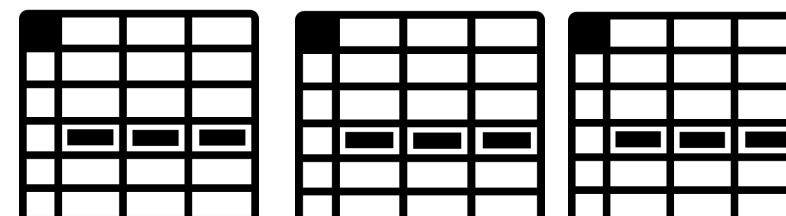
amazon



What is SQL?



Related Tables



Sales

Products

Stores

Why do we have to use SQL?

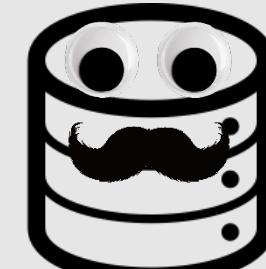


COLLECT BILLIONS AND
BILLIONS OF ROWS OF
DATA....a day.

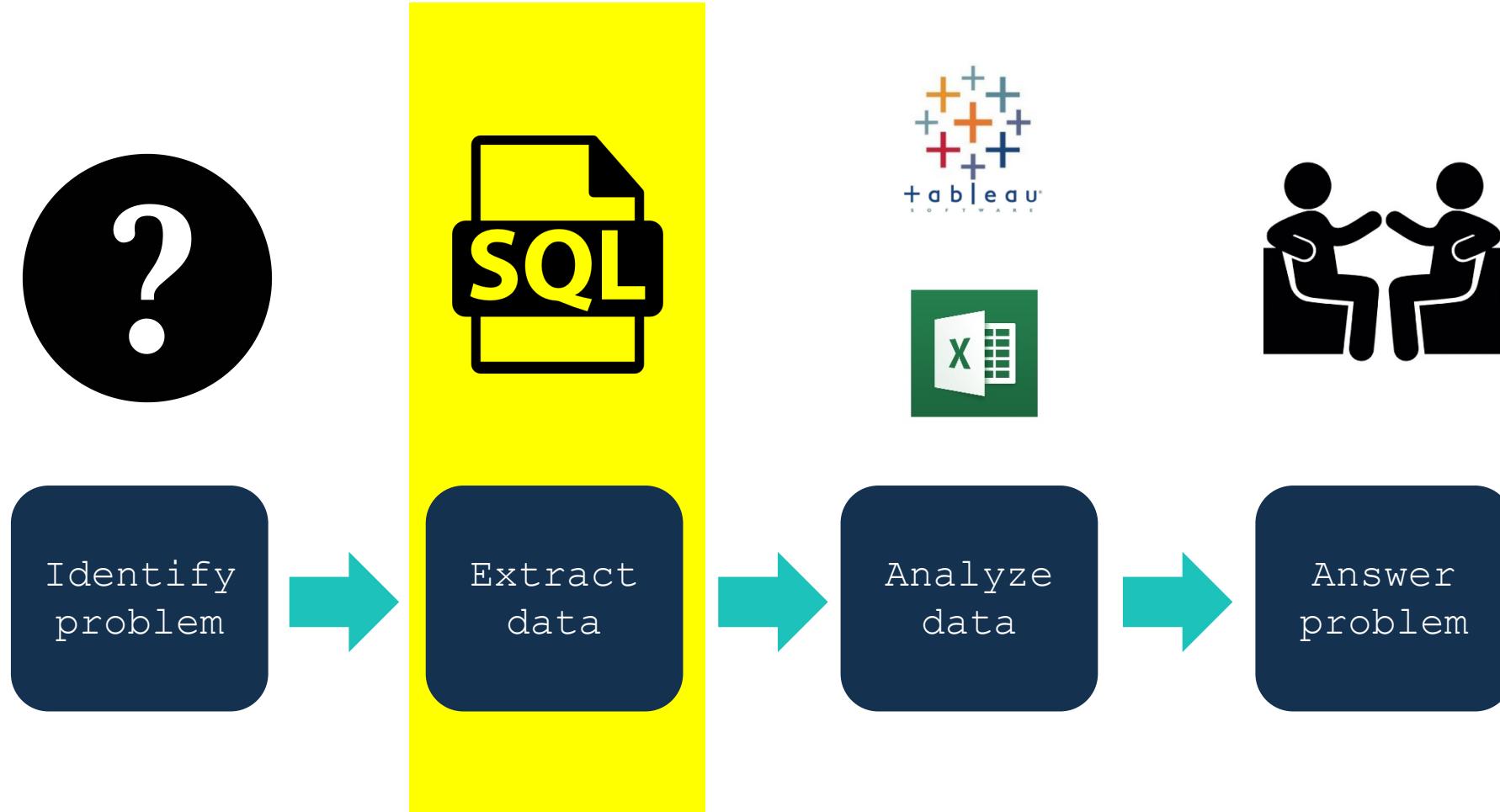
Excel Workbook
ROW Limit = 1 Million

CED .5 OZ	Ben E Keith	2 - FROZEN FOOD	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -
CAESAR CREAMY	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -
E LIQUID OLEO	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -
DUFFLE CLEAR	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -
SL-SLOTTED	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -
16OZ 161616	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -
Ben E Keith	4 - GROCERY	512 fl oz	0	0	0.00	\$ -	1.00	\$ 5.79	0.00	\$ -	0.00	\$ -
pared	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -
RMESAN SHRED	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	1.00	\$ 13.
SHELL MED USDA AA	Ben E Keith	1 - PRODUCE	0	0	0.00	\$ -	1.00	\$ 15.89	0.00	\$ -	0.00	\$ -
PPL CIDER 40GRAIN	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	1.00	\$ 17.77	0.00	\$ -
Ben E Keith	1 - PRODUCE	12/ct	0	0	0.00	\$ -	2.00	\$ 8.99	0.00	\$ -	0.00	\$ -
ICED W/GREEN CHILES	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	1.00	\$ 18.88	0.00	\$ -	0.00	\$ -
Vanilla Cr 3 Gal	Ben E Keith	6 - DAIRY	384 fl oz	0	0.00	\$ -	0.00	\$ -	0.00	\$ -	0.00	\$ -
ANCY 3% SOLIDS	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	1.00	\$ 20.69	0.00	\$ -	0.00	\$ -
M WHITE SMALL BUTTON	Ben E Keith	1 - PRODUCE	0	0	0.00	\$ -	1.00	\$ 20.98	0.00	\$ -	0.00	\$ -
SEASONED HOMESTYLE	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	1.00	\$ 22.30	0.00	\$ -
ISIANA RED HOT	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	1.00	\$ 11.24	0.00	\$ -	1.00	\$ 11.
on Celess W/Root	Ben E Keith	1 - PRODUCE	32 oz	0	0.00	\$ -	1.00	\$ 8.29	1.00	\$ 8.29	0.00	\$ -
JWN LIGHT IN BAGS	Ben E Keith	4 - GROCERY	0	0	0.00	\$ -	0.00	\$ -	1.00	\$ 27.69	0.00	\$ -
zw Jumbo	Ben E Keith	1 - PRODUCE	800 oz	0	0.00	\$ -	0.00	\$ -	1.00	\$ 13.99	0.00	\$ -

Database
ROW Limit = hold my beer



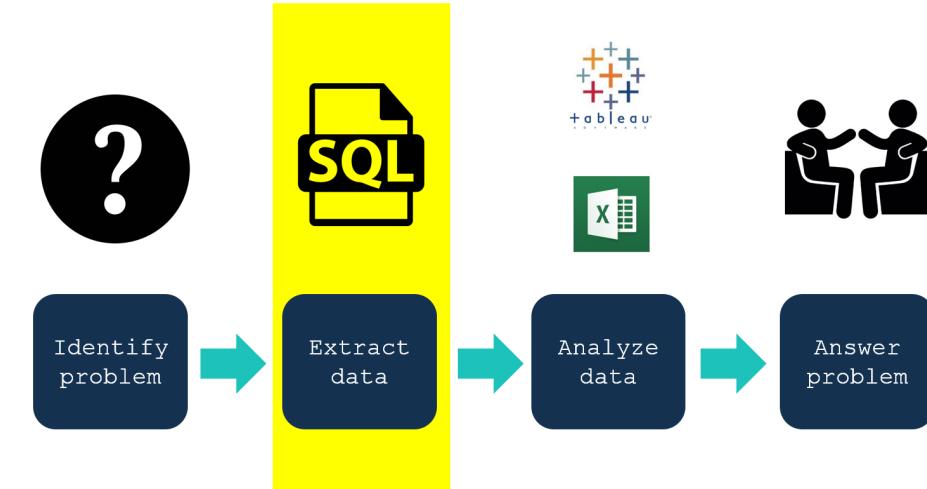
Where does SQL fit in the data flow?



What will I gain from learning SQL?



Speak to analysts in their own language



Eliminate the analytics bottleneck

Today's Agenda

INTRODUCTION (10 min)	10:00 – 10:10
Part 1: Database Fundamentals (10 min)	10:10 – 10:20
Part 2: Datatype Fundamentals (10 min)	10:20 – 10:30
Part 3: Connecting to an RDB (15 min)	10:30 – 10:45
Part 4: The Basic Syntax of SQL	11:00 – 11:30
Part 5: Filtering with WHERE	11:30 – 12:00
Part 6: Aggregations and GROUP BY	12:00 – 12:30
LUNCH – (1 HOUR)	12:30 – 01:30
Part 7: Fix the SQL CODE	01:30 – 02:15
Part 8: Joining Tables	02:15 – 03:15
Part 9: Advanced SQL Concepts	03:15 – 03:30
Part 10: Conclusion and Next Steps	03:30 – 03:45
Q:A	03:45 – 04:00

Part 1

DATABASE FUNDAMENTALS

How is data organized in a Relational Database?

Unorganized CDs



Shelf



Organized CDs



Unorganized Data

1	Californication	36319	Rock	4	10.99	€
2	By the Way	37446	Rock	5	9.99	€
3	College Dropout	38027	Rap	4	0	12.99
4	Late Registration	38594	Rap	5	2.99	0.01
5	39336	250	Rap	5	0	C
6	36641	75	Rock	0	1	Papa Roach
7	40071	40	Rap	1	0	Kid Cudi
8	38097	60	Electronic	0	1	Ratatat
9	38951	400	Electronic	0	1	Ratatat
10	Sonic Firestorm	38118	Rock	1	1	Dragonforce
11	Summer Mix	(null)	(null)	0	0	(null)
12	Party Mix	(null)	(null)	0	0	(null)
13	4	11.99	3.2	0	(null)	Common
14	3.5	11.99	4	(null)	0	T.I.
15	5	10.99	5	1	(null)	Children of Bodom



Table Schema

ALBUMS SCHEMA	
field/column	data type
row	integer
artist	text
album	text
release date	date
genre	text
plays	integer
rating	numeric(10,2)
org price	money
market value	numeric(10,2)
burned	boolean
playable	boolean



Organized Data

row	artist	album	release_date
1	Red Hot Chili Peppers	Californication	1999-06-08
2	Red Hot Chili Peppers	By the Way	2002-07-09
3	Kanye West	College Dropout	2004-02-10
4	Kanye West	Late Registration	2005-08-30
5	Kanye West	Graduation	2007-09-11
6	Papa Roach	Infest	2000-04-25
7	Kid Cudi	Man on the Moon	2009-09-15
8	Ratatat	Ratatat	2004-04-20
9	Ratatat	Classics	2006-08-22
10	Dragonforce	Sonic Firestorm	2004-05-11
11	[null]	Summer Mix 08	[null]
12	[null]	Party Mix 07	[null]
13	Common	Be	2005-05-24
14	T.I.	Paper Trail	2008-09-26
15	Children of Bodom	Children of Bodom Greatest Hits	[null]

How is data organized in a table in a Relational Database?

COLUMNS

ROWS

	row integer	artist text	album text	release_date date	genre text	plays integer	rating numeric (10,2)	org_price money	market_value numeric (10,2)	burned boolean	playable boolean
1	1	Red Hot Chili Peppers	Californication	1999-06-08	Rock	120	4.00	\$11.99	3.20	false	true
2	2	Red Hot Chili Peppers	By the Way	2002-07-09	Rock	100	3.50	\$11.99	4.00	true	false
3	3	Kanye West	College Dropout	2004-02-10	Rap	200	5.00	\$10.99	5.00	false	true
4	4	Kanye West	Late Registration	2005-08-30	Rap	300	4.00	\$9.99	7.00	false	true
5	5	Kanye West	Graduation	2007-09-11	Rap	250	4.00	\$0.00	1.75	true	true
6	6	Papa Roach	Infest	2000-04-25	Rock	75	3.50	\$11.99	0.50	false	false
7	7	Kid Cudi	Man on the Moon	2009-09-15	Rap	40	4.00	\$10.99	6.00	false	false
8	8	Ratatat	Ratatat	2004-04-20	Electro...	60	5.00	\$9.99	6.00	false	[null]
9	9	Ratatat	Classics	2006-08-22	Electro...	400	4.00	\$0.00	12.99	[null]	false
10	10	Dragonforce	Sonic Firestorm	2004-05-11	Rock	500	5.00	\$2.99	0.01	true	[null]
11	11	[null]	Summer Mix 08	[null]	[null]	1000	5.00	\$0.00	0.00	true	true
12	12	[null]	Party Mix 07	[null]	[null]	4000	5.00	\$0.00	0.00	true	true
13	13	Common	Be	2005-05-24	Rap	2000	4.50	\$0.00	15.00	true	true
14	14	T.I.	Paper Trail	2008-09-26	Rap	300	4.00	\$0.00	6.99	true	false
15	15	Children of Bodom	Children of Bodom Greate...	[null]	Metal	150	3.00	\$0.00	0.00	[null]	false

Table → ALBUMS

Part 2

DATATYPE FUNDAMENTALS

What data types are supported in SQL tables?



DATA TYPE

BOOLEAN

TEXT

INTEGER

DATE

DATA
Example

TRUE / FALSE / (NULL)
Or
1 / 0 / (NULL)

ABC / ABC123 / ABC 123
Or
(NULL)

4
Or
(NULL)

7/4/1776
Or
7/4/1776 23:00:12
Or
23:00:12 OR (NULL)

Field/Column
Example

PLAYABLE

ARTIST

RATING

RELEASE DATE

Activity 2: Guess the data type

BOOLEAN

TEXT

INTEGER

DATE

A

B

C

D

Activity 2: Guess the data type

BOOLEAN

TEXT

INTEGER

DATE

A

B

C

D

FALSE

Activity 2: Guess the data type



BOOLEAN

TEXT

INTEGER

DATE

A

B

C

D

FALSE

Country of origin: United States
Location: Minneapolis,
Minnesota
Status: Active
Formed in: 2010
Years active: 2010-present

Genre: Black Metal
Lyrical themes:
Vengeance of the
Gods, Warfare,
Ignorance of Man
Current label:
Gilead Media

Not to be confused with False from Louisville, Kentucky.



FALSE

Activity 2: Guess the data type

BOOLEAN

TEXT

INTEGER

DATE

A

B

C

D

311

Activity 2: Guess the data type

BOOLEAN

TEXT

INTEGER



DATE

A



B

C

D

311

Activity 2: Guess the data type

BOOLEAN

TEXT

INTEGER

DATE

A

B

C

D

4/29/1992

Activity 2: Guess the data type

BOOLEAN

TEXT

INTEGER

DATE

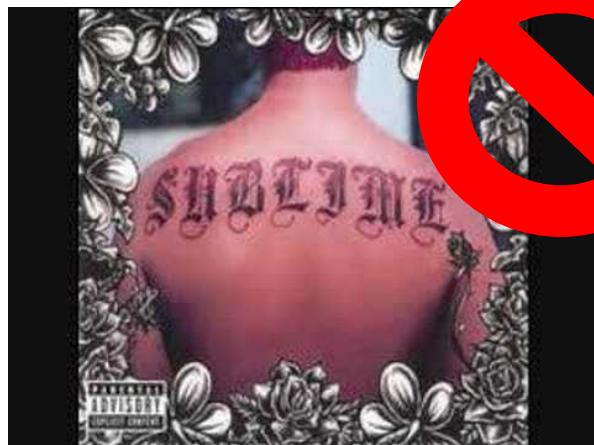


A

B

C

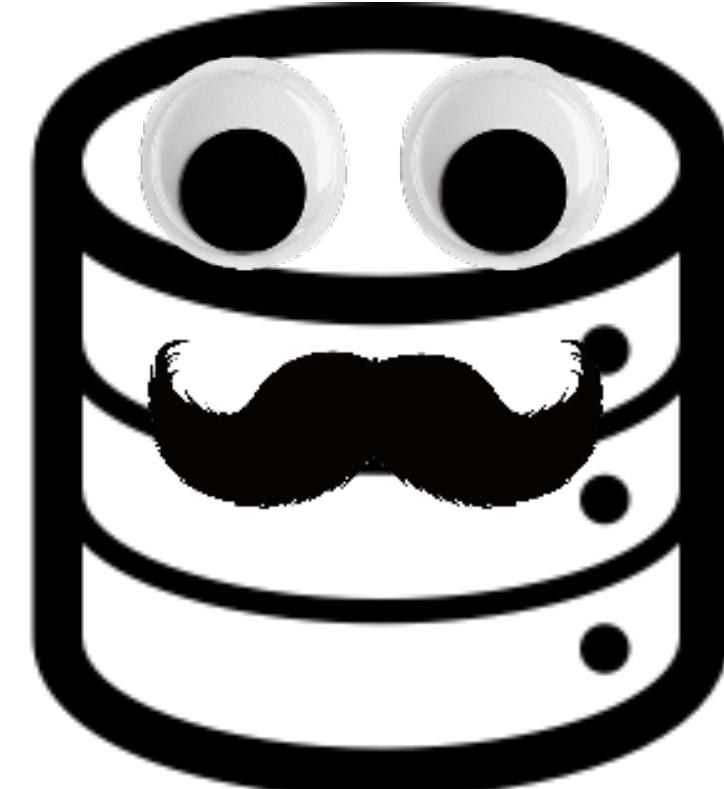
D



4/29/1992

Datatype Fundamentals: Key Takeaways

1. Appearances can be deceiving – Find the table schema first,
don't guess data types.
2. Data type has influence on how we write and format our SQL code.
3. We can write SQL code to convert datatypes too...



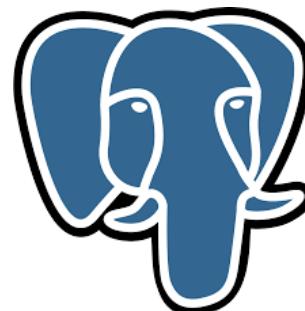
Part 3

CONNECTING TO AN RDB

How do you connect to an RDB?

THIS CLASS

Handout 2:
Connecting to
'valerianalytics'
database



YOUR ORGANIZATION



Who can help me connect to an RDB at my organization?

YOUR ORGANIZATION



ORACLE

PostgreSQL



SYBASE

1. Find the Database Administrator (DBA) at your organization. Ask them for credentials
2. If you can't find them, find someone who publishes reports. They will probably know.
3. Remember, you're probably only looking for SELECT privileges in the RDB.
4. Don't GIVE UP, hardest step is just getting access.

How to connect to valerianalytics RDB

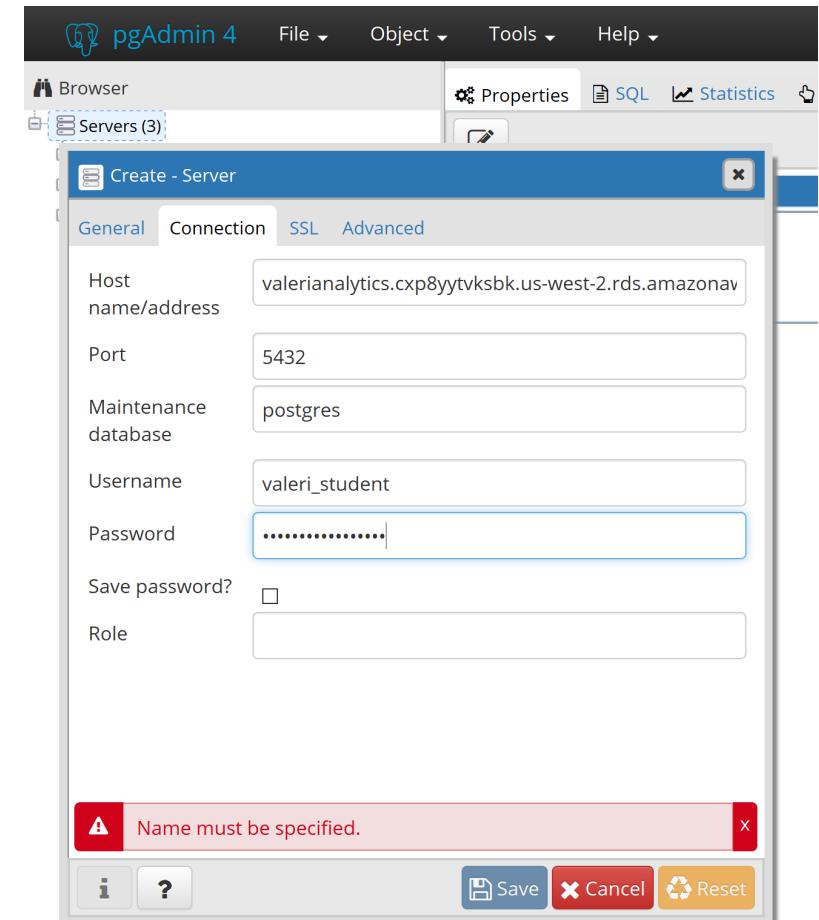
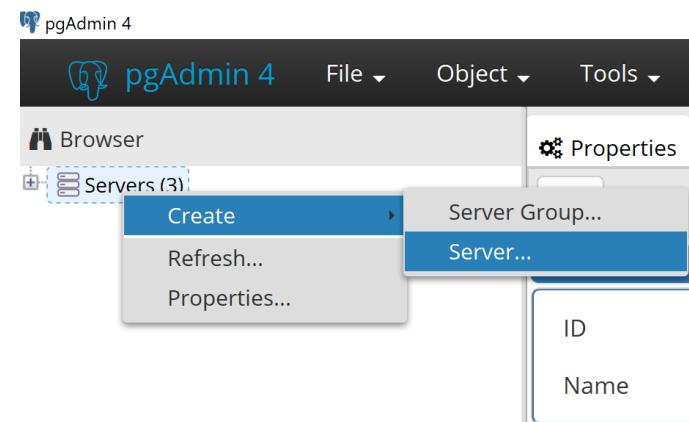
1. Download PGAdmin 4 (RDBMS)

<https://www.pgadmin.org/download/>

2. Create a server

3. Input credentials details*

(found in Handout 2: Connecting to 'valerianalytics' database)



Part 4

THE BASIC SYNTAX OF SQL

IV. How to Extract Data out of an RDB – The Basic Syntax of SQL

26

KEYWORD	WRITTEN SEQUENCE	EXECUTION SEQUENCE	WHAT IT DOES (Select A-G)	WHAT GOES THERE	EXAMPLE
SELECT	1	3	Pick an Option		
FROM	2	1			
WHERE	3	2			
GROUP BY	4	4			
HAVING	5	5			
ORDER BY	6	6			
LIMIT	7	7			

WHAT IT DOES – Options (A-G)



A	B	C	D	E	F	G
Orders results by ASC (A-Z) <u>by</u> DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table



Basic Syntax of SQL: SELECT and FROM

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (A-Z) <u>by</u> DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

TABLE

SQL Code

Result Output

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil



ID
1
2
3
4

Basic Syntax of SQL: SELECT and FROM

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT

WHAT IT DOES – Options (A-G)

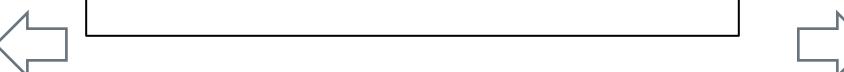
A	B	C	D	E	F	G
Orders results by ASC (<u>A-Z</u>) by DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```
SELECT
    ID
FROM TABLE
```



OR

```
SELECT ID from table
```

Result
Output

ID
1
2
3
4

Takeaways

- Tables and keywords can be written in any case
- SQL can be written on single line or multiple lines
- SELECT and FROM are required in all queries

Basic Syntax of SQL: SELECT and FROM

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT
C	E					

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (<u>A-Z</u>) by DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```
SELECT
    ID
FROM TABLE
```



OR

```
SELECT ID from table
```

Result Output

ID
1
2
3
4

Takeaways

- Tables and keywords can be written in any case
- SQL can be written on single line or multiple lines
- SELECT and FROM are required in all queries

Basic Syntax of SQL: ORDER BY

Takeaways

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT
C	E					

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (A-Z) <u>by</u> DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```
SELECT
    ID
    , ITEMNAME
FROM TABLE
ORDER BY
    VALUE DESC
```

Result Output

ID	ItemName
2	Cheese Tray
1	Red Wine
4	Knife
3	Glass of Water

Basic Syntax of SQL: ORDER BY

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT
C	E			A		

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (<u>A-Z</u>) by DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

Takeaways

- If you SELECT more than 1 column separate with a comma
- Just because you ORDER BY a column, doesn't mean you've SELECTed it

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```
SELECT
    ID
    , ITEMNAME
    , VALUE
FROM TABLE
ORDER BY
    VALUE DESC
```

Result Output

ID	ItemName	Value
2	Cheese Tray	75
1	Red Wine	50
4	Knife	20
3	Glass of Water	(null)

Basic Syntax of SQL: LIMIT

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT
C	E			A		

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (A-Z) <u>by</u> DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```

SELECT
    ID
    , ITEMNAME
    , VALUE
FROM TABLE
ORDER BY
    VALUE DESC
LIMIT 3
  
```

Result Output

ID	ItemName	Value
2	Cheese Tray	75
1	Red Wine	50
4	Knife	20

Basic Syntax of SQL: LIMIT

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT
C	E			A		B

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (<u>A-Z</u>) by DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```

SELECT
    ID
    , ITEMNAME
    , VALUE
FROM TABLE
ORDER BY
    VALUE DESC
LIMIT 3
  
```

Result Output

ID	ItemName	Value
2	Cheese Tray	75
1	Red Wine	50
4	Knife	20

Takeaways

- Limit returns a sample of rows back from the table
- LIMIT can be used with ORDER BY to rank records

Basic Syntax of SQL: WHERE

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT
C	E			A		B

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (A-Z) <u>by</u> DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```

SELECT
    ITEMNAME
    , VALUE
FROM TABLE
WHERE Category = 'Drink'

ORDER BY
    VALUE DESC
  
```

Result Output

ItemName	Value
Red Wine	50
Glass of Water	(null)

Basic Syntax of SQL: WHERE

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT
C	E	G		A		B

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (A-Z) by DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```

SELECT
  ITEMNAME
 , VALUE
 , CATEGORY
FROM TABLE
WHERE Category = 'Drink'

ORDER BY
  VALUE DESC
  
```

Result Output

ItemName	Value	Category
Red Wine	75	Drink
Glass of Water	(null)	Drink

ItemName	Value	Category
Red Wine	75	Drink
Glass of Water	(null)	Drink

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

←

→

Basic Syntax of SQL: GROUP BY

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT
C	E	G		A		B

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (A-Z) <u>by</u> DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```

SELECT
Count (ITEMNAME) AS ITEMS
,CATEGORY
FROM TABLE
GROUP BY
CATEGORY
    
```

Result Output

Items	Category
2	Drink
1	Food
1	Utensil

Basic Syntax of SQL: GROUP BY

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT
C	E	G	F	A		B

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (<u>A-Z</u>) by DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

Takeaways

- GROUP BY is used with aggregate functions (SUM,MIN,MAX,Avg,COUNT)
- Aggregate functions never go in the GROUP BY clause
- Columns can be renamed with AS

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```

SELECT
Count (ITEMNAME) AS ITEMS
,CATEGORY
FROM TABLE
GROUP BY
CATEGORY
    
```



Result Output

Items	Category
2	Drink
1	Food
1	Utensil



Basic Syntax of SQL: HAVING

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT
C	E	G	F	A		B

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (A-Z) <u>by</u> DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```

SELECT
Count (ITEMNAME) AS ITEMS
,CATEGORY
FROM TABLE
GROUP BY
CATEGORY
HAVING
COUNT (ITEMNAME) > 1
  
```

Result Output

Items	Category
2	Drink

Basic Syntax of SQL: HAVING

SELECT	FROM	WHERE	GROUP BY	ORDER BY	HAVING	LIMIT
C	E	G	F	A	D	B

WHAT IT DOES – Options (A-G)

A	B	C	D	E	F	G
Orders results by ASC (A-Z) by DESC (Z-A) order	Limits rows returned	Shows columns we want to see	Filters on aggregates	Table(s) we want to pull from	Dimensions (columns) we want aggregates GROUPED BY	Filters rows in table

TABLE

ID	ItemName	Value	Category
1	Red Wine	50	Drink
2	Cheese Tray	75	Food
3	Glass of Water	(null)	Drink
4	Knife	20	Utensil

SQL Code

```

SELECT
Count (ITEMNAME) AS ITEMS
,CATEGORY
FROM TABLE
GROUP BY
CATEGORY
HAVING
COUNT (ITEMNAME) > 1
    
```

Takeaways

- HAVING allows you to filter on aggregate columns
- WHERE allows you to filter on dimension columns
- The difference between WHERE and HAVING is the most common SQL interview question

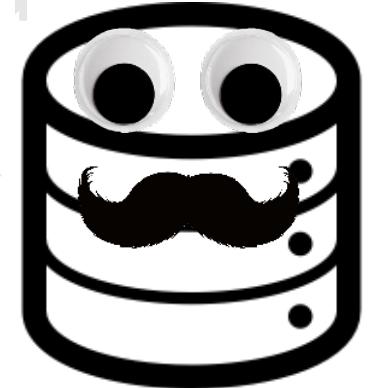
Result Output

Items	Category
2	Drink

Guided Workshop: Basic SQL Syntax



row	artist	album	release_date	genre	plays	rating	org_price	market_value	burned	playable
integer	text	text	date	text	integer	numeric(10,2)	money	numeric(10,2)	boolean	boolean
1	1 Red Hot Chili Peppers	Californication	1999-06-08	Rock	120	4.00	\$11.99	3.20	false	true
2	2 Red Hot Chili Peppers	By the Way	2002-07-09	Rock	100	3.50	\$11.99	4.00	true	false
3	3 Kanye West	College Dropout	2004-02-10	Rap	200	5.00	\$10.99	5.00	false	true
4	4 Kanye West	Late Registration	2005-08-30	Rap	300	4.00	\$9.99	7.00	false	true
5	5 Kanye West	Graduation	2007-09-11	Rap	250	4.00	\$0.00	1.75	true	true
6	6 Papa Roach	Infernal	2000-04-25	Rock	75	3.50	\$11.99	0.50	false	false
7	7 Kid Cudi	Man on the Moon	2009-09-15	Rap	40	4.00	\$10.99	6.00	false	false
8	8 Ratatat	Ratatat	2004-04-20	Electro...	60	5.00	\$9.99	6.00	false	[null]
9	9 Ratatat	Classical	2006-08-22	Electro...	400	4.00	\$0.00	12.99	[null]	false
10	10 Dragonforce	Sonic Firestorm	2004-05-11	Rock	500	5.00	\$2.99	0.01	true	[null]
11	11 [null]	Summer Mix 08	[null]	[null]	1000	5.00	\$0.00	0.00	true	true
12	12 [null]	Party Mix 07	[null]	[null]	4000	5.00	\$0.00	0.00	true	true
13	13 Common	Be	2005-05-24	Rap	2000	4.50	\$0.00	15.00	true	true
14	14 T.I.	Paper Trail	2008-09-26	Rap	300	4.00	\$0.00	6.99	true	false
15	15 Children of Bodom	Children of Bodom Greate...	[null]	Metal	150	3.00	\$0.00	0.00	[null]	false



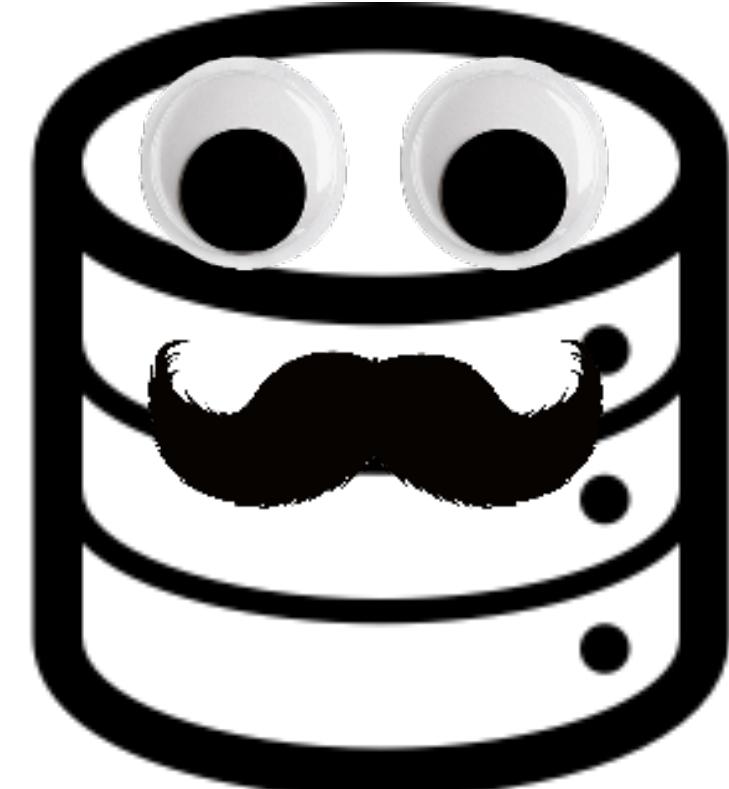
ASK A
QUESTION



GET AN
ANSWER

BASIC SQL Syntax: Key Takeaways

1. Any line, any case; SQL can be written on multiple lines or single line in any case for keywords.
2. Comma chameleon; Watch out for extra or missing commas in SELECT statement. Most common error in SQL
3. If you're having trouble combining multiple/columns fields, you might have to use CAST() to convert data types.



Part 5

FILTERING WITH WHERE

Filtering with WHERE: Outside SQL

amazon prime

Televisions television

Deliver to Michael
Seattle 98121

Browsing History Michael's Amazon.com Today's Deals Gift Cards

Televisions & Video Televisions Blu-ray Players Streaming Media Players Home Theater Systems

24 of over 2,000 results for Electronics : Television & Video : Televisions : "television"

Delivery Day

- Get It by Tomorrow

Amazon Prime

- prime
- prime | FREE One-Day

Electronics Device Model Year

- 2018
- 2017
- 2016
- 2015
- 2014

TV Display Size

- 32 Inches & Under
- 33 to 43 Inches
- 44 to 49 Inches
- 50 to 59 Inches
- 60 to 69 Inches
- 70 Inches & Up

Television Resolution

- 4K
- 1080p
- 720p

Brand

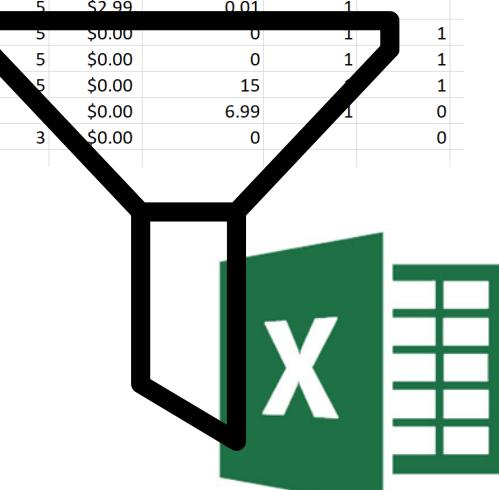
- TCL
- Samsung
- Sceptre
- Element
- LG
- Avera
- Toshiba
- SunbriteTV
- Panasonic
- VIZIO
- Sony
- Jensen
- SHARP

[See more](#)



A

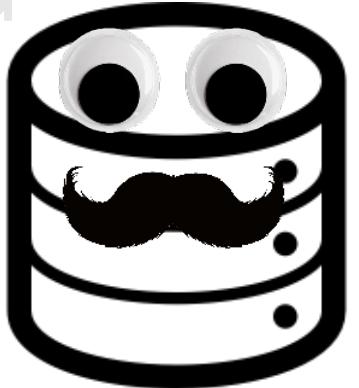
Row	Artist	Album	Release_Da	Genre	Plays	Rating	Org_Pric	Market_Val	Burned	Playable
1	Iron Maiden	Iron Maiden	6/8/1999	Rock	120	4	\$11.99	3.2	0	1
	Linkin Park	Replay	7/9/2002	Rock	100	3.5	\$11.99	4	1	0
	Linkin Park	Music Dropout	2/10/2004	Rap	200	5	\$10.99	5	0	1
	Linkin Park	Registration	8/30/2005	Rap	300	4	\$9.99	7	0	1
	Linkin Park	Hybrid Theory	9/11/2007	Rap	250	4	\$0.00	1.75	1	1
	Linkin Park	Minutes to Midnight	4/25/2000	Rock	75	3.5	\$11.99	0.5	0	0
	Linkin Park	On the Moon	9/15/2009	Rap	40	4	\$10.99	6	0	0
	Linkin Park	Minutes to Midnight	4/20/2004	Electronic	60	5	\$9.99	6	0	0
	Linkin Park	Minutes to Midnight	8/22/2006	Electronic	400	4	\$0.00	12.99	0	0
	DragonForce	Firestorm	5/11/2004	Rock	500	5	\$2.99	0.01	1	1
	DragonForce	Power Mix 08			1000	5	\$0.00	0	1	1
	DragonForce	Power Mix 07			4000	5	\$0.00	0	1	1
	DragonForce	Power Trail	5/24/2005	Rap	2000	5	\$0.00	15	1	1
	DragonForce	DragonForce Greatest Hits	9/26/2008	Rap	300	3	\$0.00	6.99	1	0
	DragonForce	DragonForce Greatest Hits		Metal	150	3	\$0.00	0	0	0



Independent Workshop: Filtering with WHERE



row	artist	album	release_date	genre	plays	rating	org_price	market_value	burned	playable
integer	text	text	date	text	integer	numeric(10,2)	money	numeric(10,2)	boolean	boolean
1	1 Red Hot Chili Peppers	Californication	1999-06-08	Rock	120	4.00	\$11.99	3.20	false	true
2	2 Red Hot Chili Peppers	By the Way	2002-07-09	Rock	100	3.50	\$11.99	4.00	true	false
3	3 Kanye West	College Dropout	2004-02-10	Rap	200	5.00	\$10.99	5.00	false	true
4	4 Kanye West	Late Registration	2005-08-30	Rap	300	4.00	\$9.99	7.00	false	true
5	5 Kanye West	Graduation	2007-09-11	Rap	250	4.00	\$0.00	1.75	true	true
6	6 Papa Roach	Infernal	2000-04-25	Rock	75	3.50	\$11.99	0.50	false	false
7	7 Kid Cudi	Man on the Moon	2009-09-15	Rap	40	4.00	\$10.99	6.00	false	false
8	8 Ratatat	Ratatat	2004-04-20	Electro...	60	5.00	\$9.99	6.00	false	[null]
9	9 Ratatat	Classics	2006-08-22	Electro...	400	4.00	\$0.00	12.99	[null]	false
10	10 Dragonforce	Sonic Firestorm	2004-05-11	Rock	500	5.00	\$2.99	0.01	true	[null]
11	11 [null]	Summer Mix 08	[null]	[null]	1000	5.00	\$0.00	0.00	true	true
12	12 [null]	Party Mix 07	[null]	[null]	4000	5.00	\$0.00	0.00	true	true
13	13 Common	Be	2005-05-24	Rap	2000	4.50	\$0.00	15.00	true	true
14	14 T.I.	Paper Trail	2008-09-26	Rap	300	4.00	\$0.00	6.99	true	false
15	15 Children of Bodom	Children of Bodom Greate...	[null]	Metal	150	3.00	\$0.00	0.00	[null]	false



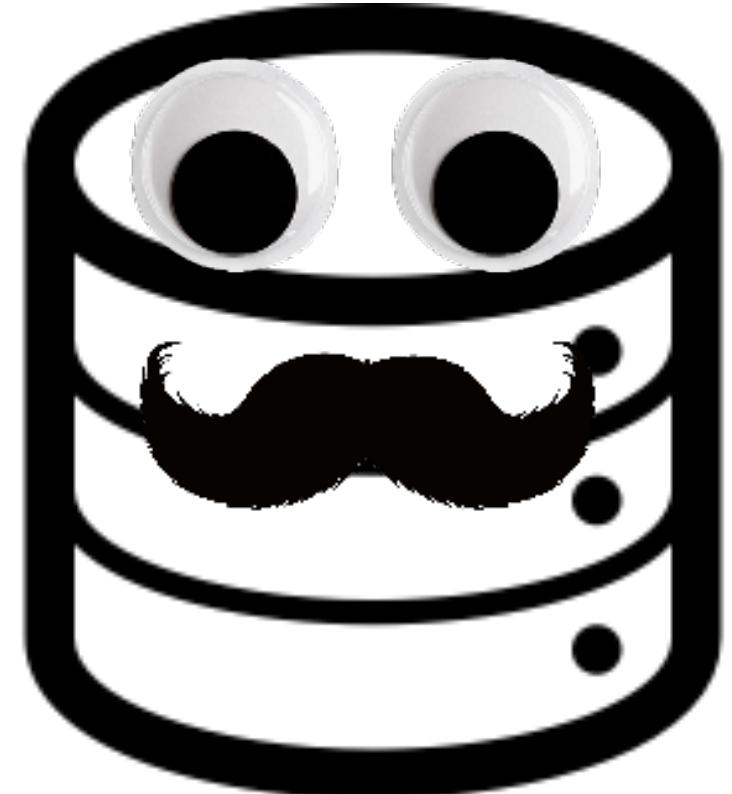
ASK A
QUESTION



GET AN
ANSWER

Filtering with WHERE: Key Takeaways

1. Many nuances; Syntax for filtering depends on the column data type you are working with.
2. You don't need to SELECT columns you are filtering on in the WHERE clause
3. Use TO_CHAR() function to simplify dates when filtering. Use LIKE with '%' to look for patterns.



Part 6

AGGREGATIONS AND GROUP BY

Aggregations and Group BY: An aggregation without a Group BY

Pie Chart A: Total Plays

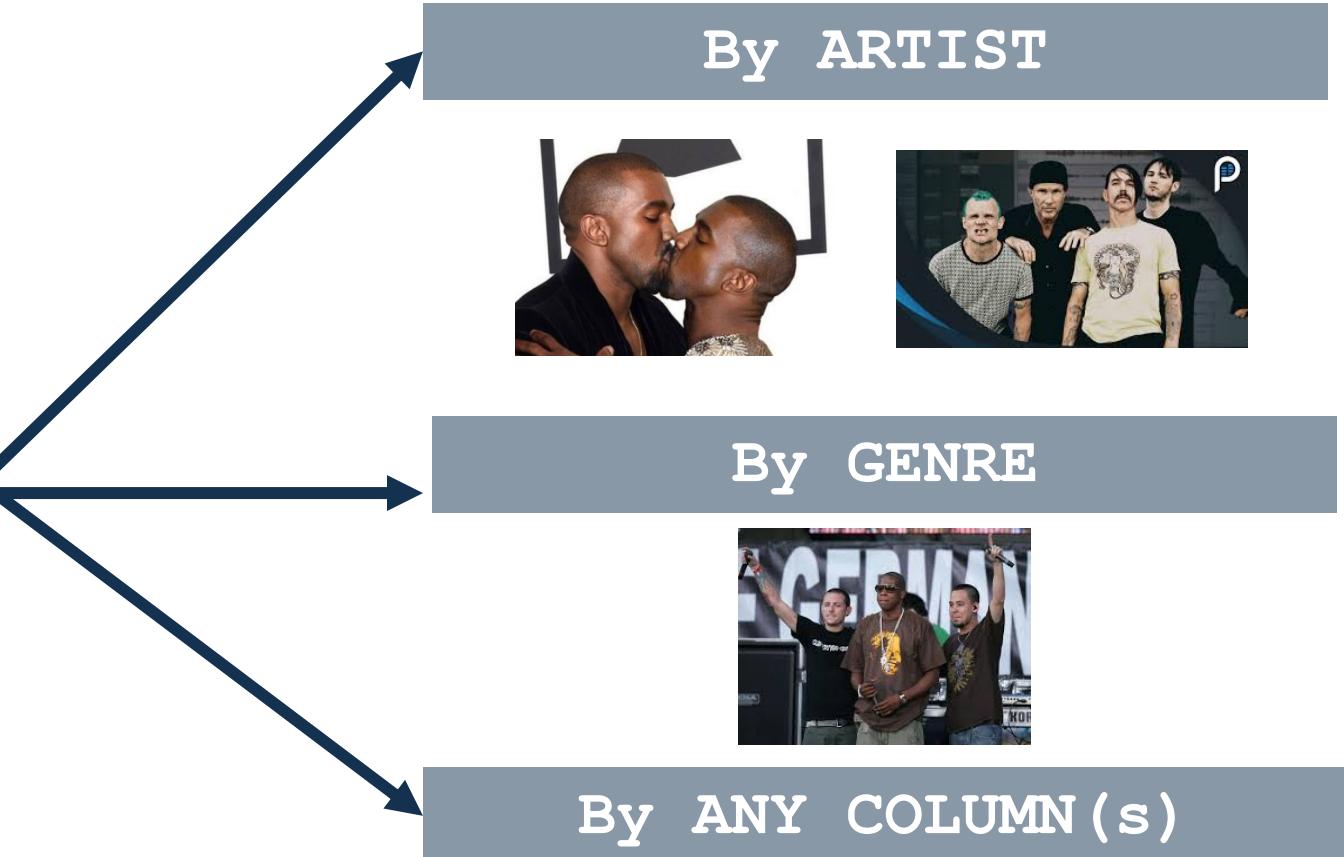
9,495
plays



Great analytics
foundation.

Aggregations and Group BY: What can we group our aggregations by?

9,495 plays



Aggregations and Group BY: Like a Pivot Table in Excel

By ARTIST

PivotTable Fields

Choose fields to add to report:

Search

- Row
- Artist
- Album
- Release_Date
- Genre
- Plays
- Rating
- Org_Price
- Market_Value
- Burned
- Playable

More Tables...

Drag fields between areas below:

Filters	Columns
Rows	Values
Artist	Sum of Plays

Row Labels	Sum of Plays
(blank)	5000
Common	2000
Kanye West	750
Dragonforce	500
Ratatat	460
T.I.	300
Red Hot Chili Peppers	220
Children of Bodom	150
Papa Roach	75
Kid Cudi	40
Grand Total	9495

By GENRE

PivotTable Fields

Choose fields to add to report:

Search

- Row
- Artist
- Album
- Release_Date
- Genre
- Plays
- Rating
- Org_Price
- Market_Value
- Burned
- Playable

More Tables...

Drag fields between areas below:

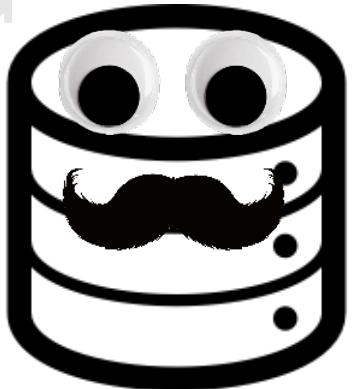
Filters	Columns
Rows	Values
Genre	Sum of Plays

Row Labels	Sum of Plays
Electronic	460
Metal	150
Rap	3090
Rock	795
(blank)	5000
Grand Total	9495

Guided Workshop: Aggregations and GROUP BY



row	integer	artist	text	album	text	release_date	genre	plays	rating	numeric(10,2)	org_price	market_value	numeric(10,2)	burned	boolean	playable	boolean
1	1	Red Hot Chili Peppers		Californication		1999-06-08	Rock	120	4.00	\$11.99		3.20	false	true			
2	2	Red Hot Chili Peppers		By the Way		2002-07-09	Rock	100	3.50	\$11.99		4.00	true	false			
3	3	Kanye West		College Dropout		2004-02-10	Rap	200	5.00	\$10.99		5.00	false	true			
4	4	Kanye West		Late Registration		2005-08-30	Rap	300	4.00	\$9.99		7.00	false	true			
5	5	Kanye West		Graduation		2007-09-11	Rap	250	4.00	\$0.00		1.75	true	true			
6	6	Papa Roach		Infernal		2000-04-25	Rock	75	3.50	\$11.99		0.50	false	false			
7	7	Kid Cudi		Man on the Moon		2009-09-15	Rap	40	4.00	\$10.99		6.00	false	false			
8	8	Ratatat		Ratatat		2004-04-20	Electro...	60	5.00	\$9.99		6.00	false	[null]			
9	9	Ratatat		Classics		2006-08-22	Electro...	400	4.00	\$0.00		12.99	[null]	false			
10	10	Dragonforce		Sonic Firestorm		2004-05-11	Rock	500	5.00	\$2.99		0.01	true	[null]			
11	11	[null]		Summer Mix 08		[null]	[null]	1000	5.00	\$0.00		0.00	true	true			
12	12	[null]		Party Mix 07		[null]	[null]	4000	5.00	\$0.00		0.00	true	true			
13	13	Common		Be		2005-05-24	Rap	2000	4.50	\$0.00		15.00	true	true			
14	14	T.I.		Paper Trail		2008-09-26	Rap	300	4.00	\$0.00		6.99	true	false			
15	15	Children of Bodom		Children of Bodom Greate...		[null]	Metal	150	3.00	\$0.00		0.00	[null]	false			



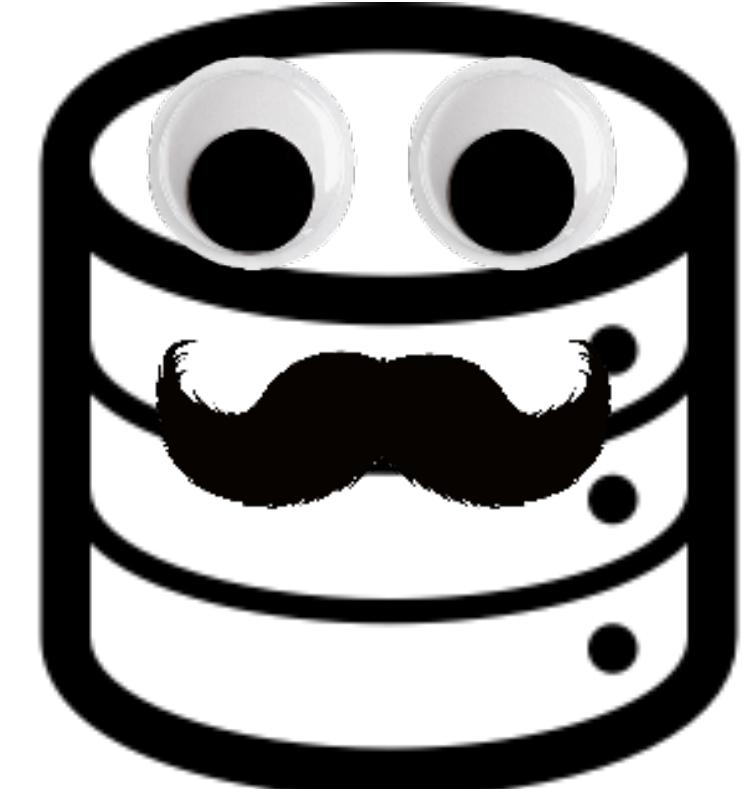
ASK A
QUESTION



GET AN
ANSWER

Aggregations and GROUP BY: Key Takeaways

1. Use GROUP BY when you want to understand aggregates (SUM/ COUNT/ etc.) by specific fields/columns.
I.e. Total sales by Country/Channel.
2. Aggregate functions and aliases, **DO NOT** go in the GROUP BY.
3. You **CAN NOT** filter on aggregate functions in the WHERE clause, can only filter in HAVING clause.
4. You **CAN NOT** perform on aggregate on an aggregate (i.e. Avg(Sum(Plays)) by Genre, need to perform in two steps (more on this later).



LUNCH

(1 Hour)

Return at 1:30 PM

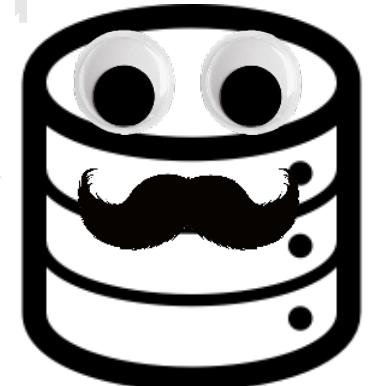
Part 7

FIX THE CODE

GROUP Workshop 4: Fix the Code



row	integer	artist	text	album	text	release_date	genre	plays	rating	numeric(10,2)	org_price	market_value	numeric(10,2)	burned	boolean	playable	boolean
1	1	Red Hot Chili Peppers		Californication		1999-06-08	Rock	120	4.00	\$11.99		3.20	false	true		false	
2	2	Red Hot Chili Peppers		By the Way		2002-07-09	Rock	100	3.50	\$11.99		4.00	true	false			
3	3	Kanye West		College Dropout		2004-02-10	Rap	200	5.00	\$10.99		5.00	false	true			
4	4	Kanye West		Late Registration		2005-08-30	Rap	300	4.00	\$9.99		7.00	false	true			
5	5	Kanye West		Graduation		2007-09-11	Rap	250	4.00	\$0.00		1.75	true	true			
6	6	Papa Roach		Infest		2000-04-25	Rock	75	3.50	\$11.99		0.50	false	false			
7	7	Kid Cudi		Man on the Moon		2009-09-15	Rap	40	4.00	\$10.99		6.00	false	false			
8	8	Ratatat		Ratatat		2004-04-20	Electro...	60	5.00	\$9.99		6.00	false	[null]		false	
9	9	Ratatat		Classics		2006-08-22	Electro...	400	4.00	\$0.00		12.99	[null]	false			
10	10	Dragonforce		Sonic Firestorm		2004-05-11	Rock	500	5.00	\$2.99		0.01	true	[null]			
11	11	[null]		Summer Mix 08		[null]	[null]	1000	5.00	\$0.00		0.00	true	true			
12	12	[null]		Party Mix 07		[null]	[null]	4000	5.00	\$0.00		0.00	true	true			
13	13	Common		Be		2005-05-24	Rap	2000	4.50	\$0.00		15.00	true	true			
14	14	T.I.		Paper Trail		2008-09-26	Rap	300	4.00	\$0.00		6.99	true	false			
15	15	Children of Bodom		Children of Bodom Greate...		[null]	Metal	150	3.00	\$0.00		0.00	[null]	false			



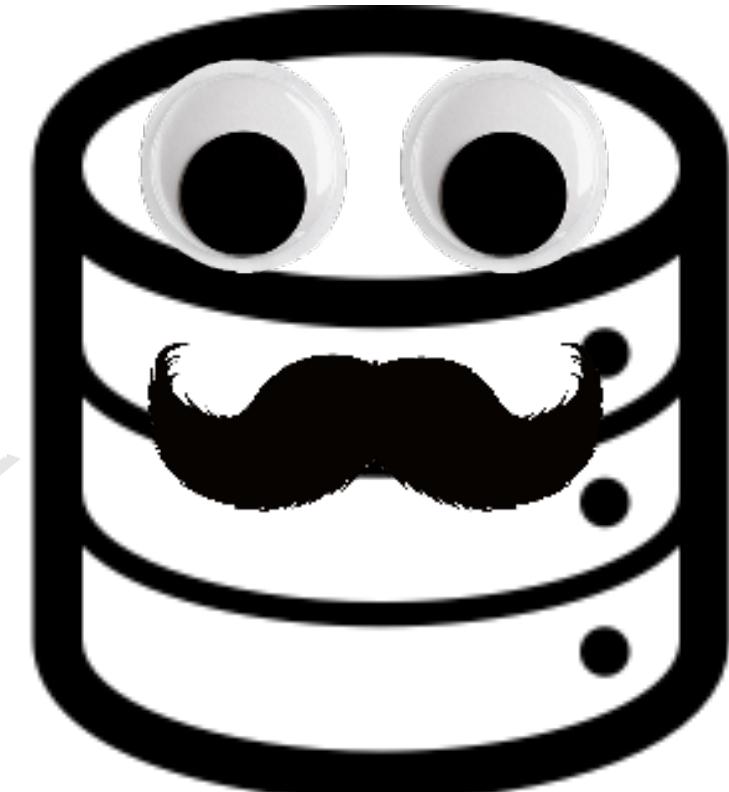
ASK A
QUESTION



GET AN
ANSWER

Fix the Code: Key Takeaways

1. **Work backwards;** draw the result-output first
2. **Remove and add back;** Eliminate what doesn't work in the code by commenting out lines with '---' and '/* */'
3. **Ask for help;** collaborate with others



Part 8

JOINING TABLES

What is a JOIN?

Table A - Orders	
OrderID	CustomerID
12345	1
12346	2
12344	8
12355	NULL



Table B - Customers	
CustomerID	CustomerName
1	Chuckie. F
2	Tommy. P
3	Angelica P.
4	Dr. Liptshitz

Why use a JOIN? – We have a question we can't answer from a single table.

What is a JOIN?

Table A - Orders	
OrderID	CustomerID
12345	1
12346	2
12344	8
12355	NULL



LEFT/PRIMARY
TABLE

Table B - Customers	
CustomerID	CustomerName
1	Chuckie. F
2	Tommy. P
3	Angelica P.
4	Dr. Liptshitz

RIGHT/SECONDARY
TABLE

JOINED TABLE		
OrderID	CustomerID	CustomerName
BLAH	BLAH	BLAH
BLAH	BLAH	BLAH
BLAH	BLAH	BLAH

Same thing as a
VLOOKUP in Excel



What is required to join tables in SQL?

Table A - Orders		Table B - Customers	
OrderID	CustomerID	CustomerID	CustomerName
12345	1	1	Chuckie. F
12346	2	2	Tommy. P
12344	8	3	Angelica P.
12355	NULL	4	Dr. Liptshitz

1. At least 2 tables, with at least 1 column/field in common
2. Column/field(s) tables have in common, **MUST** be same data type
3. The column/field in Table B **should** be unique, otherwise duplication will occur (*more on this later*)

What is the JOIN syntax?

```

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

FROM
ORDERS A

INNER JOIN CUSTOMERS B

ON B.CUSTOMERID = A.CUSTOMERID
  
```

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)

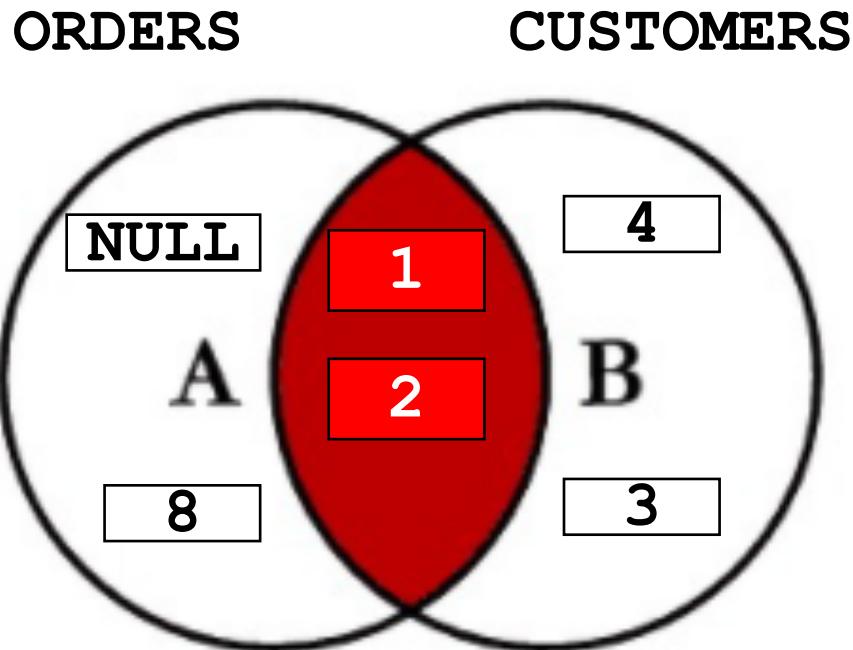
3. You need indicate what **STYLE of JOIN** (*more than 1 style*)

Table A - Orders	
OrderID	CustomerID
12345	1
12346	2
12344	8
12355	NULL

Table B - Customers	
CustomerID	CustomerName
1	Chuckie. F
2	Tommy. P
3	Angelica P.
4	Dr. Liptshitz

INNER JOIN RESULT SET (CustomerID)		
OrderID	CustomerID	CustomerName
12345	1	Chuckie. F
12346	2	Tommy. P

How do we visualize an INNER JOIN?



```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```

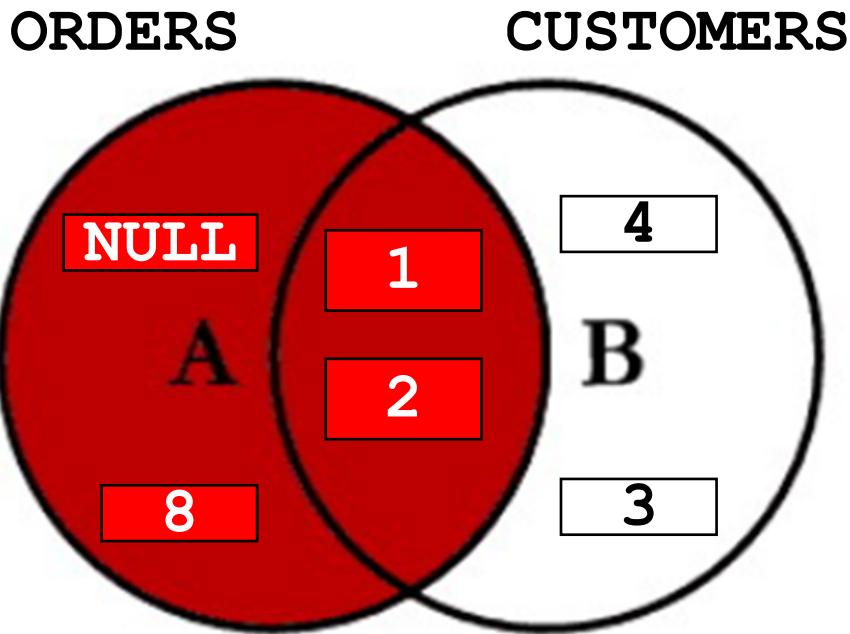
**JOIN → EVERYTHING TABLE A and B HAVE
IN COMMON**

Table A - Orders		Table B - Customers	
OrderID	CustomerID	CustomerID	CustomerName
12345	1	1	Chuckie. F
12346	2	2	Tommy. P
12344	8	3	Angelica P.
12355	NULL	4	Dr. Liptshitz

INNER JOIN RESULT SET (CustomerID)

OrderID	CustomerID	CustomerName
12345	1	Chuckie. F
12346	2	Tommy. P

How do we visualize a LEFT JOIN?



```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```

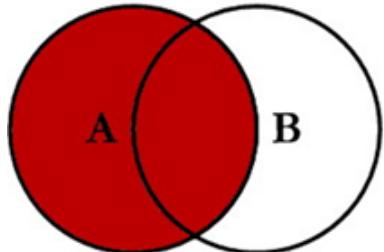
LEFT JOIN → EVERYTHING THAT's in
Table A regardless of whether it's in
Table B

Table A - Orders		Table B - Customers	
OrderID	CustomerID	CustomerID	CustomerName
12345	1	1	Chuckie. F
12346	2	2	Tommy. P
12344	8	3	Angelica P.
12355	NULL	4	Dr. Liptshitz

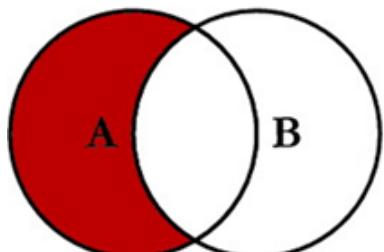
LEFT JOIN RESULT SET		
OrderID	CustomerID	CustomerName
12345	1	Chuckie. F
12346	2	Tommy. P
12344	8	NULL
12355	NULL	NULL

What are the other types of JOINS?

SQL JOINS

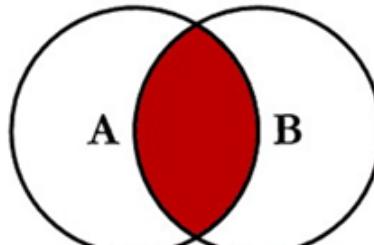


```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```

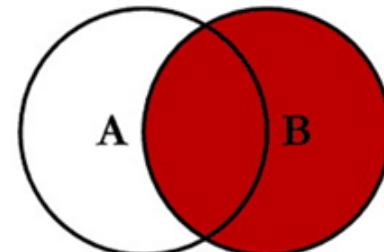


```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```

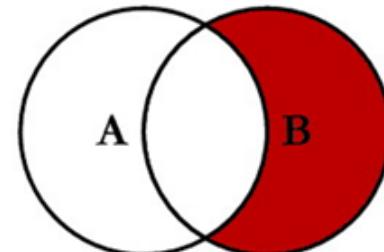
```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



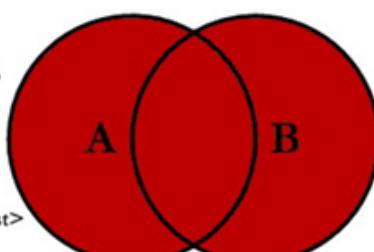
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



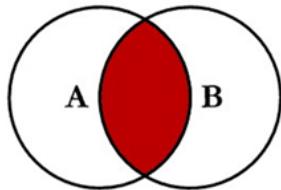
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

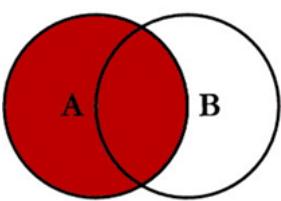
Group Activity: Which JOIN do I use? (20 min)

Table A: ORDERS	
ORDER_ID	CUST_ID
12345	567
12346	457
12366	123
14567	(null)
19453	(null)
15678	234



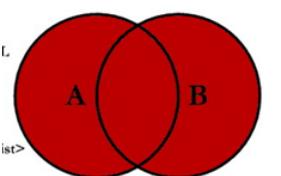
GROUP 1

Question: What is the count of customers and orders by state?



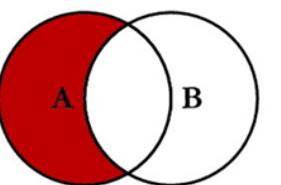
GROUP 2

Step 1: Make a joined table of order_id, cust_id, and state. Draw circles to help you.



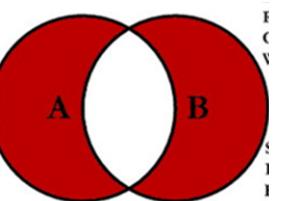
GROUP 3

Step 2: Make a table of the count of orders and customers in each state. Remember, don't count NULL customer or order IDs.



GROUP 4

Step 3: Tell us the total order and customer count overall.

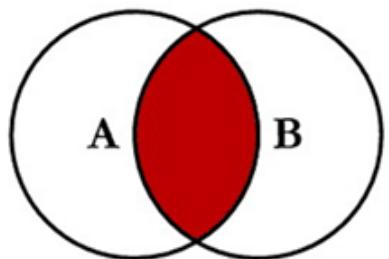


GROUP 5

Which JOIN do I use? (Most Common Use Cases)

"Give me a list of total orders/customers by customer's state."

INNER JOIN



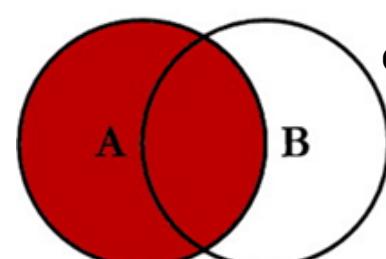
GROUP 1

```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```

WILL EXCLUDE ALL ORDERS FOR
CUSTOMERS THAT AREN'T IN TABLE B

*Consequence: Order count
underreported*

LEFT JOIN



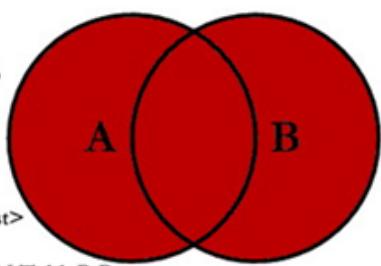
GROUP 2

```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```

WILL INCLUDE ORDERS FOR ALL
CUSTOMERS, REGARDLESS OF
EXISTENCE IN TABLE B.

*Consequence: Some orders won't
have a state assigned to them
(will appear as NULL for State)*

FULL OUTER JOIN



```
FT JOIN TableB B  
ON A.Key = B.Key  
HERE B.Key IS NULL
```

```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```

© C.L.N

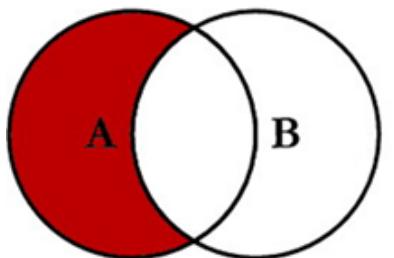
WILL INCLUDE ALL ORDERS AND CUSTOMERS
FROM BOTH TABLES

*Consequence: Some order's won't have a
state assigned to them, and some
customer's won't have an order assigned
to them.*

Which JOIN do I use? (Fringe Use Cases)

"Give me a list of total orders/customers by customer's state."

LEFT JOIN, WHERE
CUST_ID IS NULL in
TABLE B

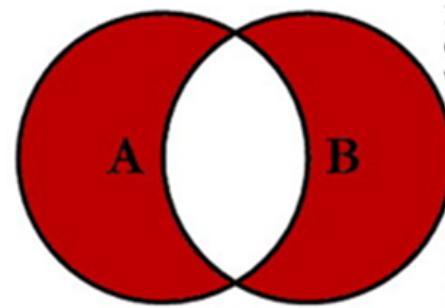


```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL.
```

USE CASE: You only want to see orders for customers that aren't in Table B.

Tell DBA to update Customer Attributes; Table B

FULL OUTER JOIN, WHERE
CUST_ID IS NULL in TABLE A
and B



RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL.

```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL.
OR B.Key IS NULL
```

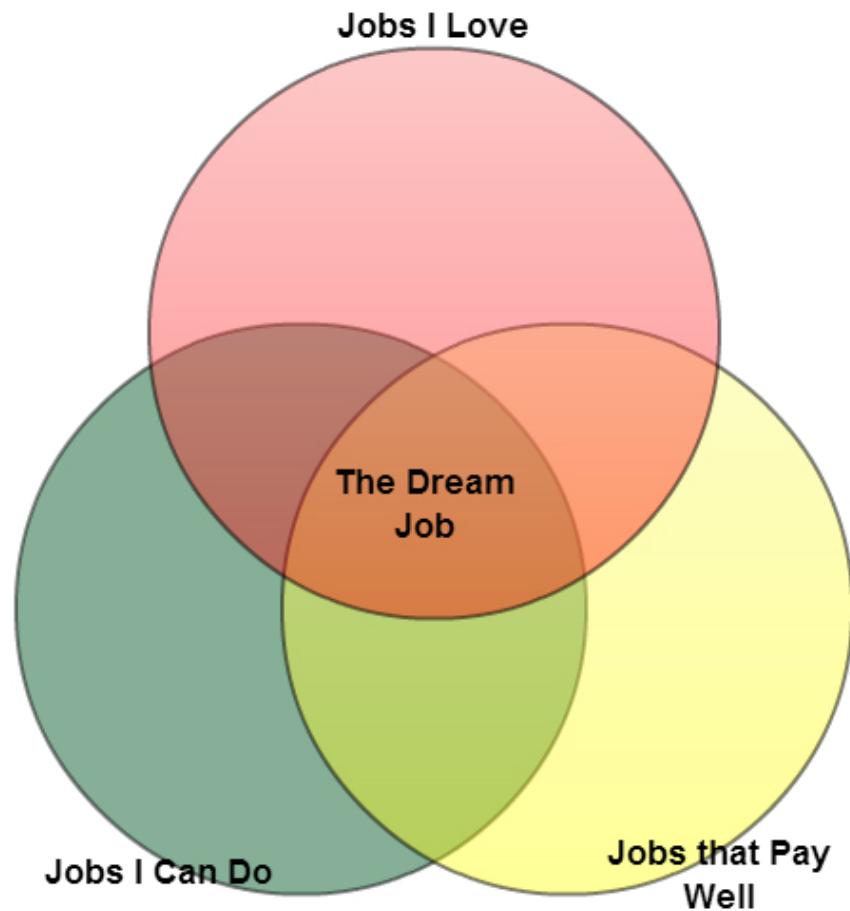
flatt, 2008

USE CASE: You only want to see orders from customers that don't exist in Table B
AND

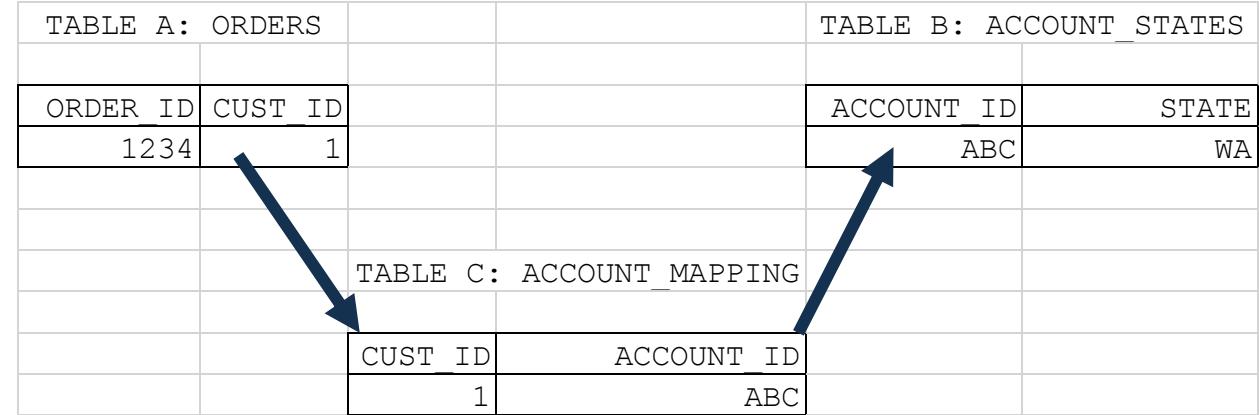
You only want to see customers that don't exist in Table A (don't have an order)

Can you join to more than 1 table?

```
SELECT c.field, a.field,  
b.field, a.field,c.field  
FROM table1 a  
JOIN table2 b  
ON a.field=b.field  
JOIN table3 c  
ON a.field=c.field
```



What if you want to join two tables that don't share the same column?



```
1  SELECT
2
3    A.ORDER_ID
4    ,A.CUST_ID
5    ,B.STATE
6
7  FROM
8
9    ORDERS A
10
11 LEFT JOIN ACCOUNT_MAPPING C ON A.CUST_ID = C.CUST_ID
12
13 LEFT JOIN ACCOUNT_STATES B ON B.ACCOUNT_ID = C.ACCOUNT_ID
```

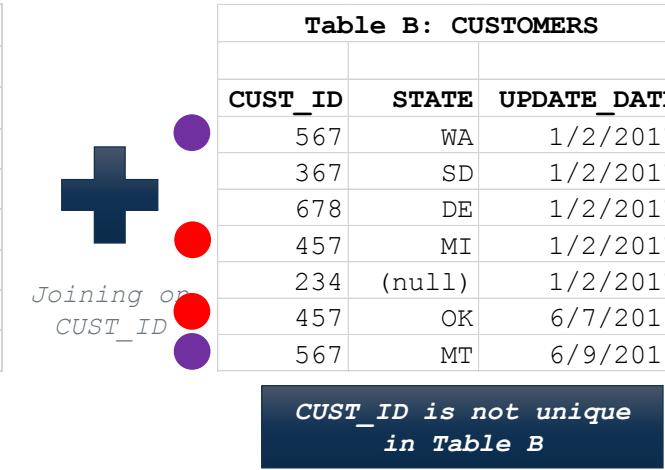
SOLUTION: DAISY
CHAINING

What if you are getting duplicate records or rows after performing a join?



REQUIREMENTS FOR A JOIN BETWEEN TWO TABLES

1. At least 2 tables, with at least 1 column/field in common
2. Column/field(s) tables have in common, **MUST** be same data type
3. The column/field in Table B **should** be unique, otherwise duplication will occur (*more on this later*)



INNER JOIN

ORDER_ID	CUST_ID	STATE
12345	567	WA
12345	567	MT
12346	457	OK
12346	457	MI

LEFT JOIN

ORDER_ID	CUST_ID	STATE
12345	567	WA
12345	567	MT
12346	457	OK
12346	457	MI
12366	123	(null)
14567	(null)	(null)
19453	(null)	(null)
15678	234	(null)

SOLUTIONS

Dedup TABLE B via Subquery or CTE.

Join on more than column, if possible.

Instructor Led: Write the SQL JOIN as a Class

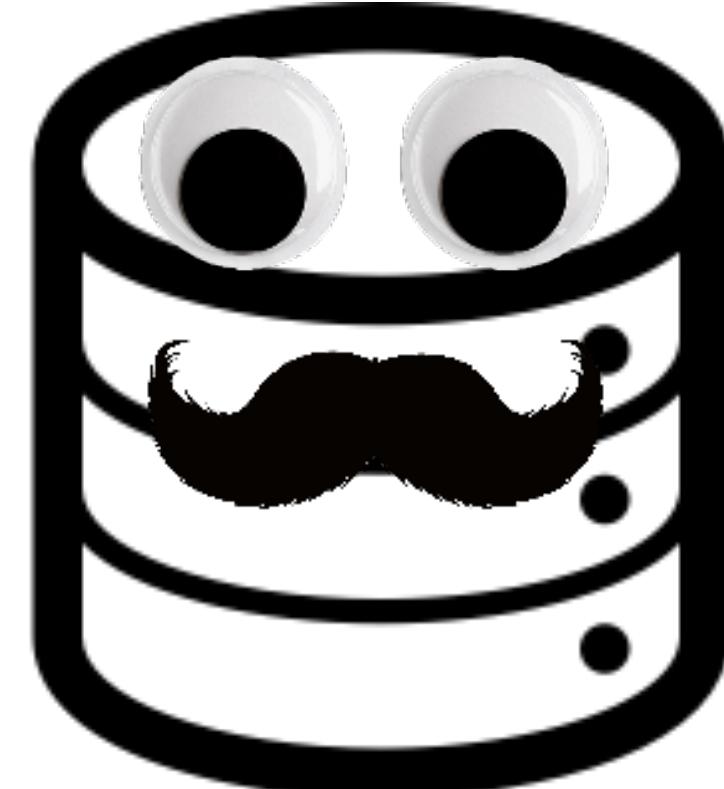
Hey Analyst, Give me the top 10 stores in terms of sales for 2014. I want them listed by store name.

Framework for approach

1. Draw the result-output first. Determine if you need a join or not.
2. Make sure 3 JOIN requirements are met.
3. Follow 3 rules of the JOIN syntax.
4. Pick appropriate JOIN. Be ready to defend.

JOINING Tables: Key Takeaways

1. If you want to join two tables, they must have 1 column in common, and of the same data type. The rows/records in Table B should be unique, otherwise duplication. Check TABLE B with supplied code.
2. The JOIN style will determine the result-output. Think about what records/rows will be excluded. Left joins are safest.
3. Work backwards, draw result output first, then determine ingredients to writing code.



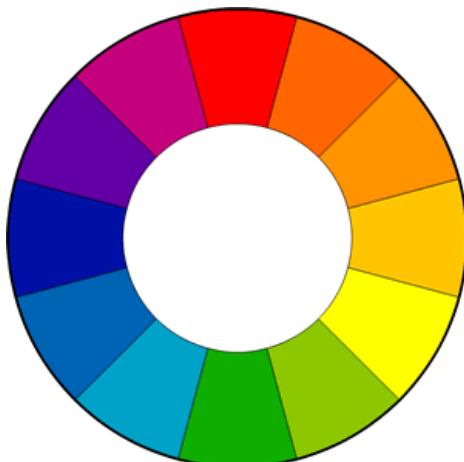
Part 9

ADVANCED SQL TOPICS

Writing Conditional Statements in SQL (CASE)

Question: Give me total sales for primary color items vs. non-primary color items...

Prod_ID	Color	Unit_Sold	Primary_Color
1	Red	3	
2	Pink	2	
3	Yellow	5	
4	Blue	3	
5	Green	1	
6	Brown	6	



IF
Statement



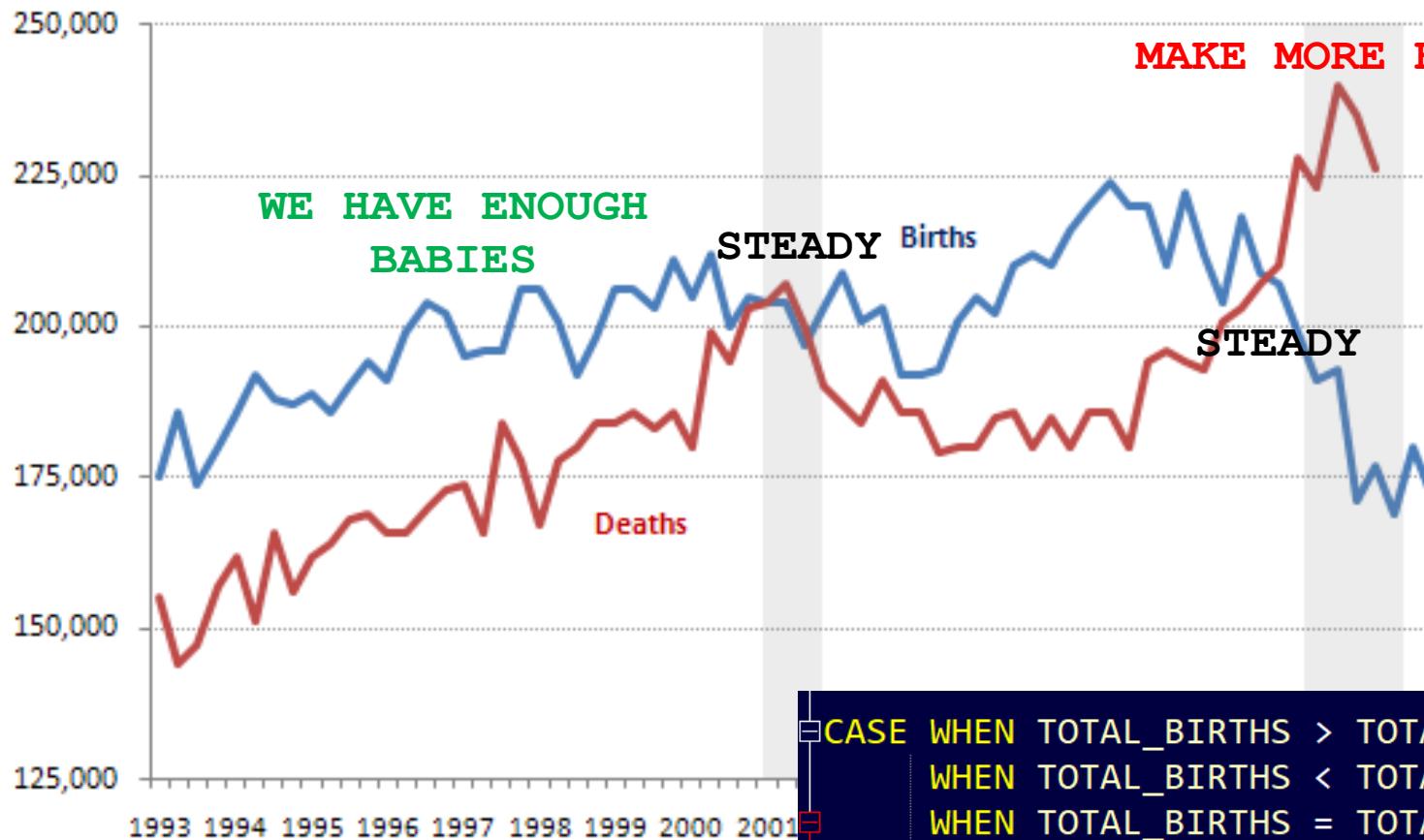
CASE
Statement

Color	Unit_Sold	Primary_Color
Red	3	=IF(OR(H21="Red", H21="Yellow", H21="Blue"), 1, 0)
Pink	2	0
Yellow	5	1
Blue	3	1
Green	1	0
Brown	6	0

```
10  SELECT
11    prod_id
12    ,color
13    ,unit_sold
14  CASE WHEN COLOR IN ('Red','Yellow','Blue')
15    THEN 1 ELSE 0 END AS PRIMARY_COLOR
16
17  FROM TABLE
```

CASE Statement Use Cases

Chart 5. Quarterly establishment births and deaths, 1993–2010



```
CASE WHEN TOTAL_BIRTHS > TOTAL_DEATHS THEN 'WE HAVE ENOUGH BABIES'  
WHEN TOTAL_BIRTHS < TOTAL_DEATHS THEN 'MAKE MORE BABIES'  
WHEN TOTAL_BIRTHS = TOTAL_DEATHS THEN 'STEADY'  
END AS POPULATION_STATUS
```

- Case statements are useful for
- Categorization
- Inspiring action

Writing Subqueries in SQL – Use Cases

- What is a subquery?

- It's a tool for performing operations in multiple steps. **It's a query within a query.**
Use to return data that will be used in the main query.

- When would you use?

- You want to take an aggregate of an aggregate
- You want to de-duplicate a table to perform a JOIN
- You want reference information from multiple tables
- You want to use **WHERE to filter on an aggregate**

RUNS
2nd

RUNS
1st

```
SELECT
    A.PLAYS
    ,A.GENRE
FROM
    (
        SELECT
            SUM(PLAYS) as PLAYS
            ,GENRE
        FROM ALBUMS
        GROUP BY GENRE) A
WHERE
    A.PLAYS > 1000
```

- What is the order of operations for a subquery?

- From inside out (**Subquery runs first → Main query runs second**)

- What are the requirements?

- Subqueries must be enclosed in ()
- An alias must be supplied to them

Writing Subqueries in SQL – Example Flow

RUNS
2nd

RUNS
1st

```
SELECT
    A.PLAYS
    ,A.GENRE

FROM

(SELECT
    SUM(PLAYS) as PLAYS
    ,GENRE
    FROM ALBUMS
    GROUP BY GENRE) A

WHERE
    A.PLAYS > 1000
```

RUNS 1st – Creation of
Table A

	plays	genre
	bigint	text
1	5000	[null]
2	795	Rock
3	150	Metal
4	3090	Rap
5	460	Electro...

RUNS 2nd – Run SELECT
STATEMENT, off Table A

	plays	genre
	bigint	text
1	5000	[null]
2	3090	Rap

Part 10

CONCLUSION

We extracted the data, using SQL. Now what do we do with it? **Remember analytics workflow...**

Hey Analyst,

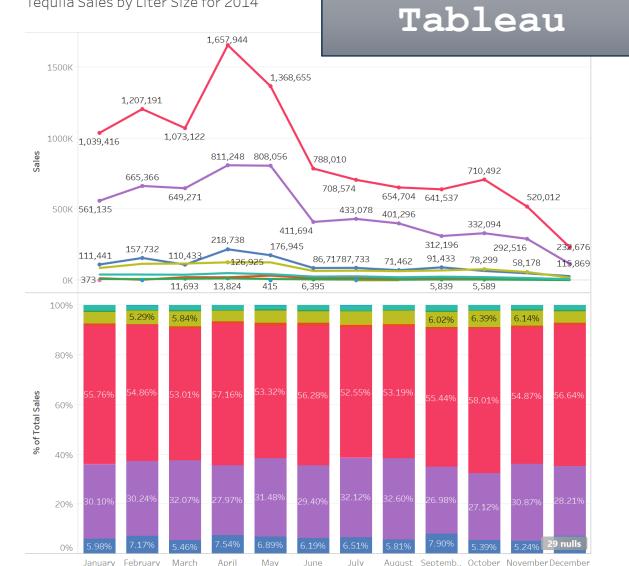
People

What were tequila sales for 2014?

What month and liter sizes had the highest sales?

SQL

```
1 SELECT
2     sum(total) as SALES
3     ,to_char(date, 'YYYY') as YEAR
4     ,category_name
5     ,liter_size
6     ,to_char(date, 'MM') as MONTH
7
8     FROM SALES
9
10    WHERE CATEGORY_NAME = 'TEQUILA'
11
12    GROUP BY
13        to_char(date, 'YYYY')
14        ,category_name
15        ,liter_size
16        ,to_char(date, 'MM')
```



Identify problem



Extract data



Analyze data



Answer problem

We offer courses on this part of the chain as well..

Hey Boss,

People

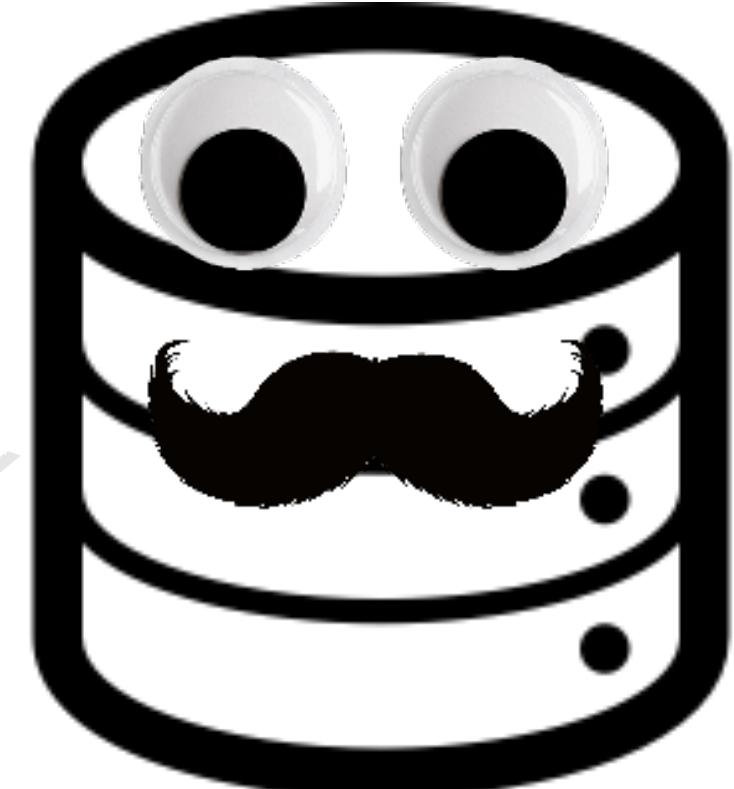
Tequila sales were \$19.3MM for 2014.

The highest sales month was Apr, with \$1.7MM in sales.

The most popular liter size was 750 ml (55% of sales)

Conclusion: Key Takeaways

1. The more you talk to a database, you more comfortable you'll get.
2. Today's class gave us a framework for not just talking to '**Red**', but to any relational database.
3. Don't let anyone stop you from getting to know a database; push hard.



Example Email to Send To Your DBA to Get Access

Hello DBA,

I'd to request SELECT access to our relational database that stores (x) data. Could you supply me the necessary credentials or Wiki for gaining access?

Could you also let me know

1. Which version of SQL our organization uses? (E.g. Oracle / MySQL / Hive / PostgreSQL / IBM BIG Insights / SQL Server / etc.)
2. What RDBMS (or GUI) we use as a query tool?
3. Any information regarding the schema structure of the tables in the database for reference.

Thanks,

Your Name

Q&A

(15 min)

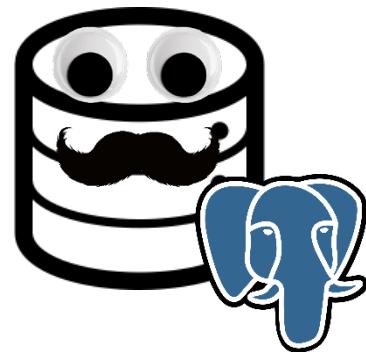
What are the other dialects of SQL databases?

PROS

Free
Scalable and can handle a lot of data

CONS

Spotty
Documentation
Configuration can be confusing



PROS

Most Popular for organizations
Robust tools, cutting edge

CONS

Expensive, for small orgs
Significant resource maintenance

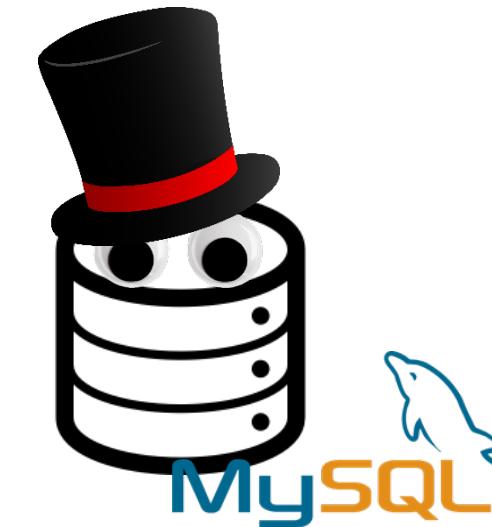


PROS

Free
Most popular for web-based apps
Can connect to Oracle

CONS

You need to pay for support

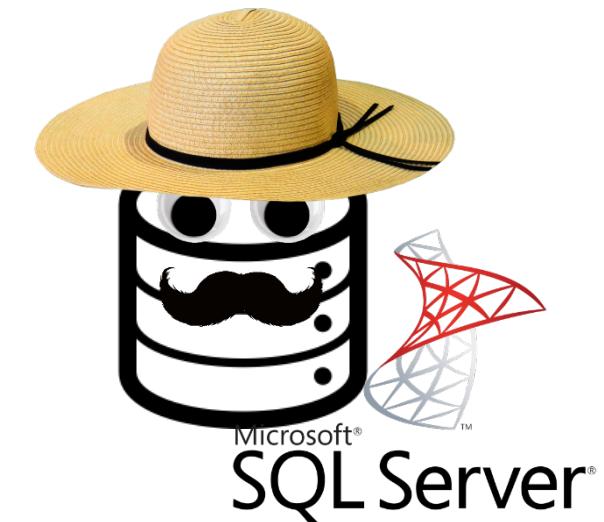


PROS

Fast and stable
Works well MS products

CONS

Expensive
Resource maintenance



How does an RDB consume raw data?



Random data sources

Raw disorganized data

Clean organized data

Database
administrators
and analysts



**Will only consume what
we tell it to consume**