
ORACLE®
ACADEMY

Solve It With SQL

Lesson 2
Introduction to SQL

ORACLE ACADEMY

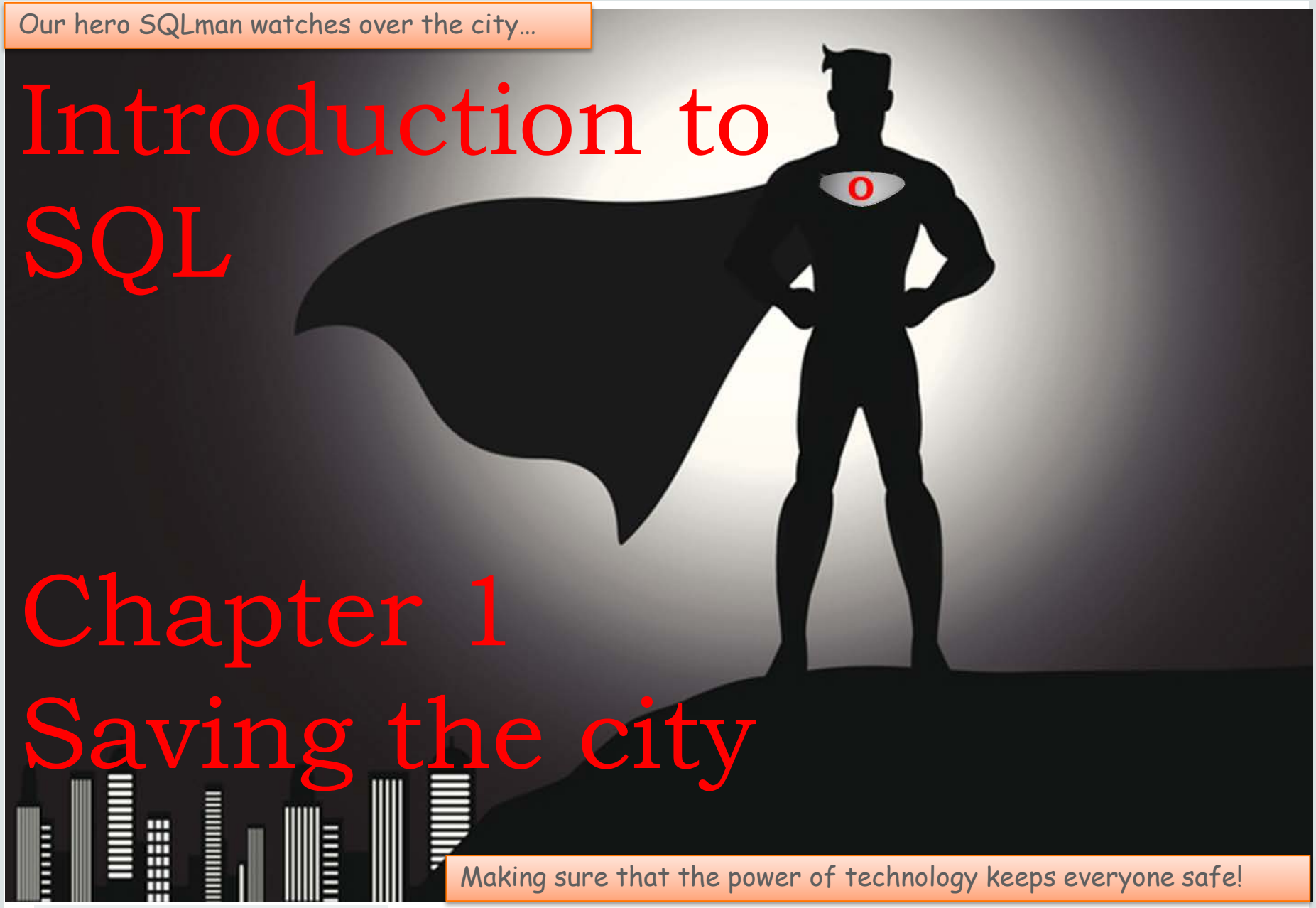
Overview

This lesson covers the following topics:

- Using APEX
- What is SQL
- SQL Statements
- Conditions
- Conditions Using Logical Operators
- Rules of Precedence
- Using Comparison Operators
- Data Manipulation Language – Insert.

Our hero SQLman watches over the city…

Introduction to SQL



Chapter 1
Saving the city

Making sure that the power of technology keeps everyone safe!

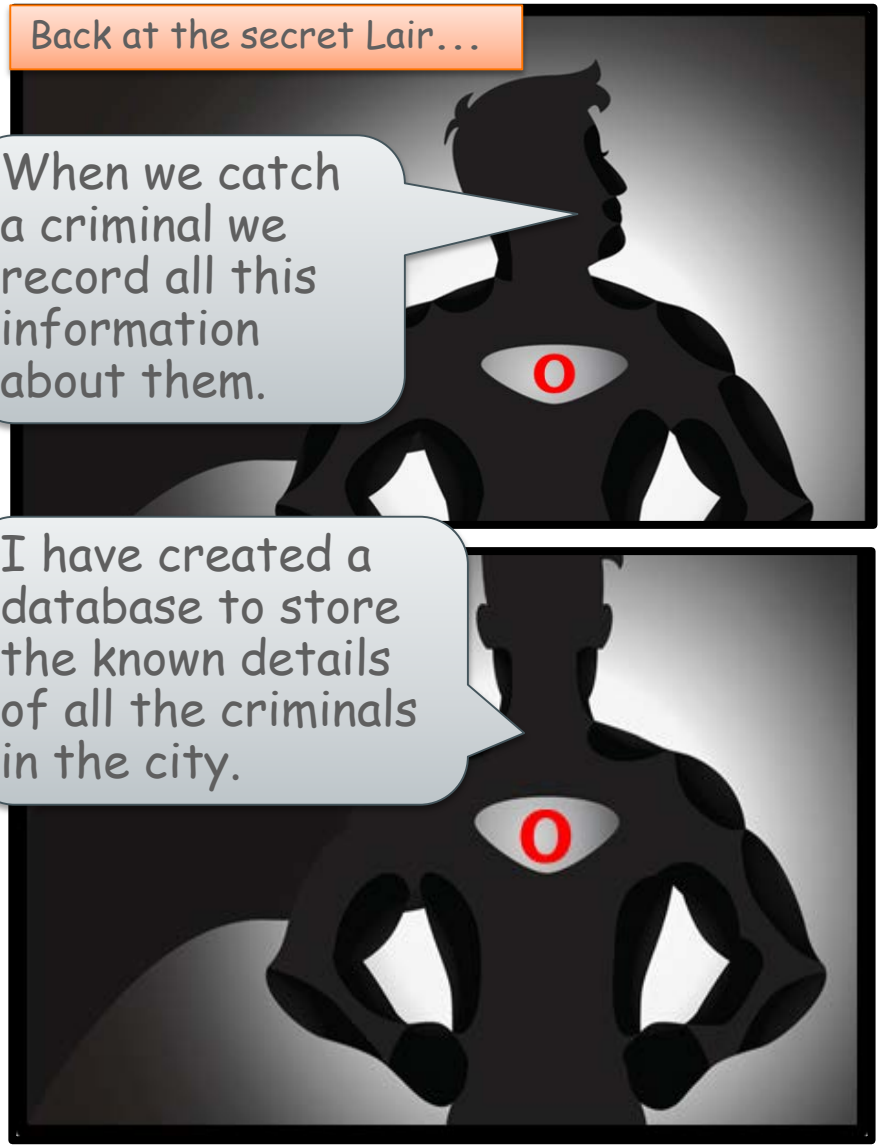
Solve It With SQL

Lesson 2
Introduction to SQL

Part 1 - What is a Database?

Ha! Ha! Ha! Stealing is fun!

Recording Criminal's Details



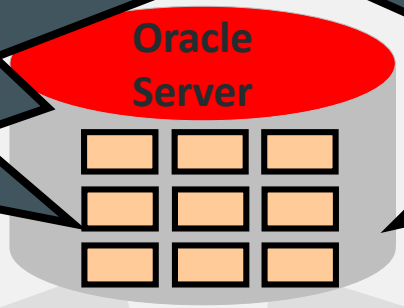
Criminal Properties:

- Name
- Sex
- Age
- Height
- Hair_Color
- Eye_Color
- Facial_Hair
- Tattoos
- Glasses
- Scars
- Feet_Size

How Is Data Organized in Databases?

Time for some theory...

Data is stored in a two-dimensional matrix known as a table.

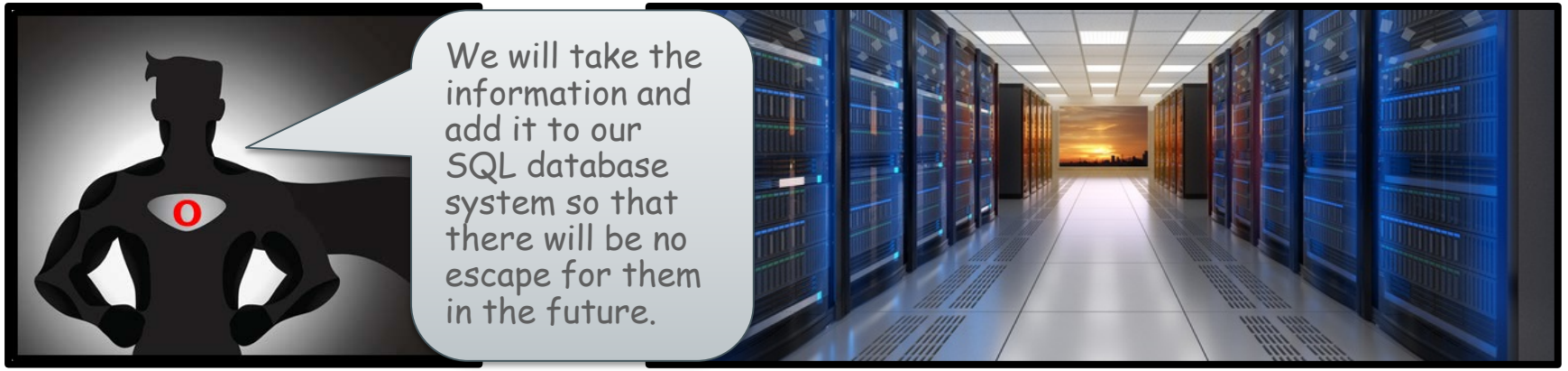


RDBMS software is used to manage reading and manipulating data.

SUSPECT_ID	NAME	SEX	AGE
210	Nasim Jennings	Female	21
211	Mia Greer	Female	78
212	Byron Underwood	Male	18
213	Madison Gilbert	Male	42

NAME	LOCATION	MEMBERS
Bouncers	Sydney	3
Unjust League	Winter Bay	6
Weegie YT	Glasgopolis	9
Sinister Seven	Falkirk	7

Suspect Examples



Name : Clayton Lara
Sex : Female
Age : 29
Height : Tall
Hair_Color : Blonde
Eye_Color : Brown
Facial_Hair : No
Tattoos : No
Glasses : No
Scars : Yes
Feet_Size : Large

Name : Josh Stone
Sex : Male
Age : 42
Height : Medium
Hair_Color : Black
Eye_Color : Blue
Facial_Hair : Yes
Tattoos : Yes
Glasses : Yes
Scars : No
Feet_Size : Medium

Suspect Properties : Types and Values

Before we can add the information to our system we need to think about how we can store that information in our database. The data type used is based on the values that will be stored in the field!

Once we have all of the data stored in the database we can then use the power of SQL to catch the thieves.

Property Name	Type	Potential Values
Name	Text	Rachel | Robyn | Jim
Sex	Text	Male | Female
Age	Numeric	18 – 85
Height	Text	Tall, Medium, Short)
Hair_Color	Text	Black | Brown | Blonde| Red
Eye_Color	Text	Blue | Brown | Green)
Facial_Hair	Text	Yes | No
Tattos	Text	Yes | No
Glasses	Text	Yes | No
Scars	Text	Yes | No
Feet_Size	Text	Large | Medium | Small

Solve It With SQL

Lesson 2
Introduction to SQL

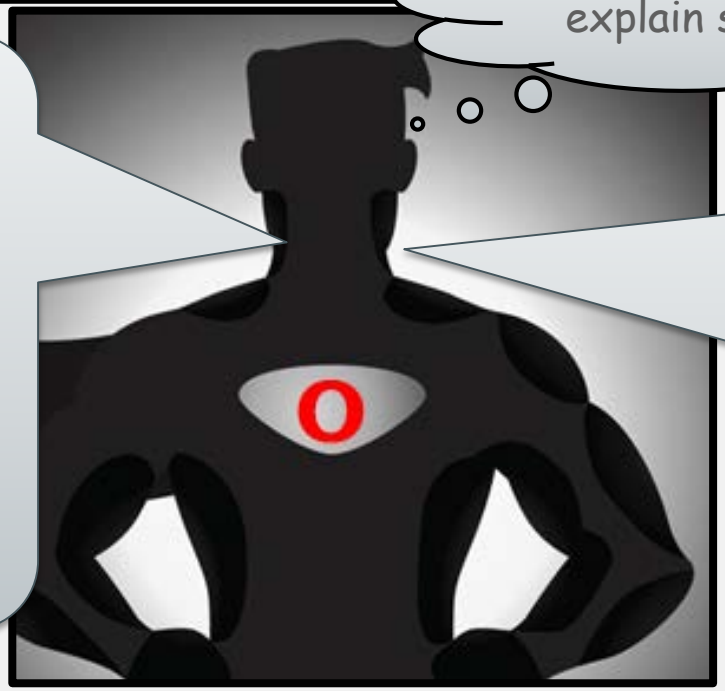
Part 2 – What is SQL?

What Is SQL?

Before we go on I should explain some of these terms!!

Structured query language (SQL) is the set-based language used to access data.

It is a set of statements with which all programs and users access data in an Oracle database.



SQL provides an interface to a relational database and provides statements that help work with the database.

With SQL you do not need to know how the data is arranged physically.

To access the database, you execute a SQL statement

Now we need a tool to learn and use the SQL that will help us catch the thieves.

For this we will use Oracle Application Express

Oracle Application Express

Build applications using only your web browser.

Solve It With SQL

Lesson 2
Introduction to SQL

Part 3 – Using SQL in APEX

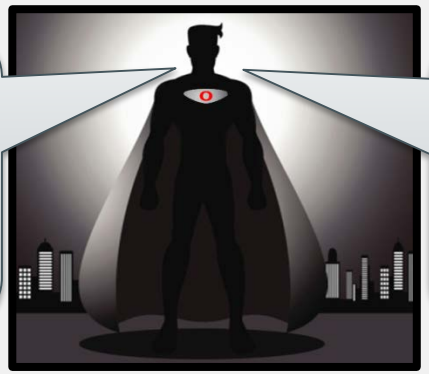
Using SQL in Application Express

In APEX we can use SQL to:

- Create, replace, alter, and drop database objects
- Insert, update and delete rows in a table
- Query data stored in the database
- Control access to the database and database objects
- Guarantee database consistency and integrity.

SQL provide benefits to all types of users:

- Application programmers
- Database administrators
- Managers
- End users

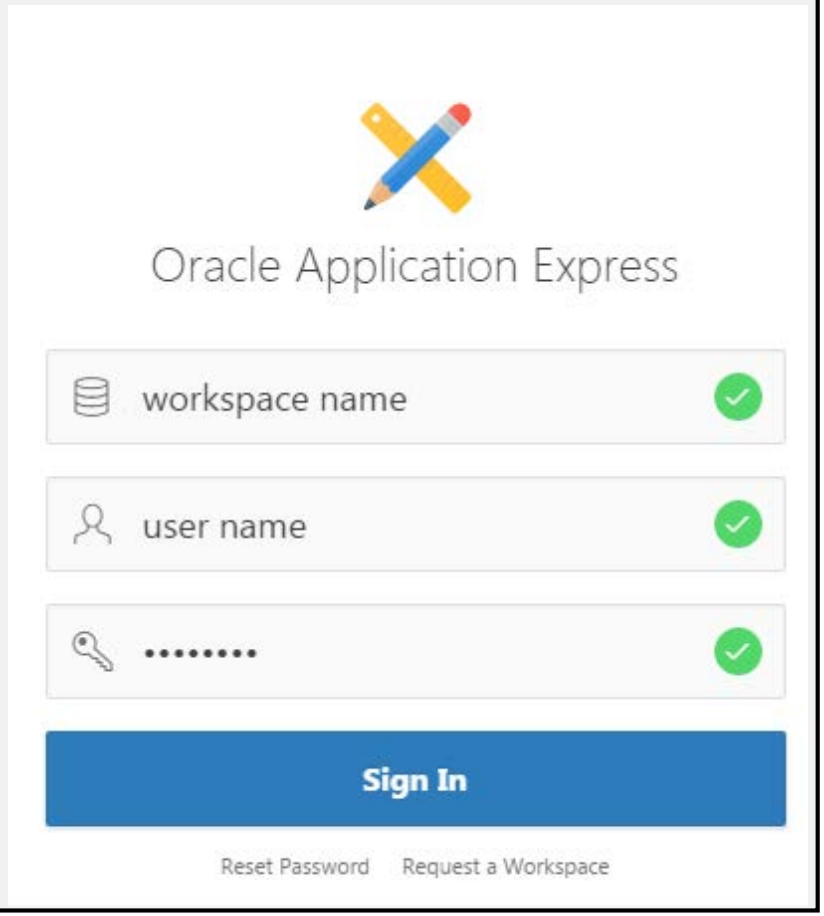


Without this I wouldn't be able to keep track of all the criminals in my city and my citizens would never be safe!!

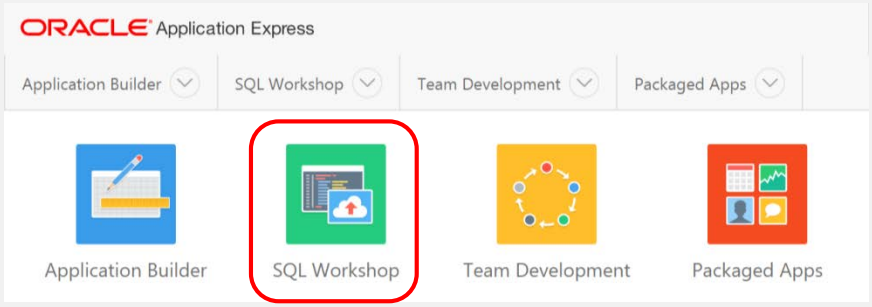
Log In to Application Express

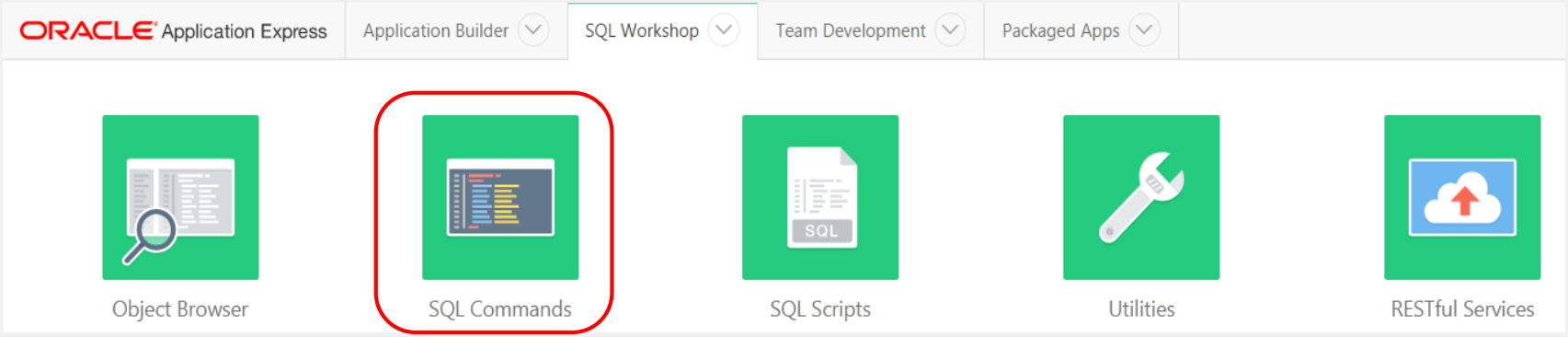
Enter the correct URL in your browser address bar

Using the login details provided sign in to your APEX workspace.

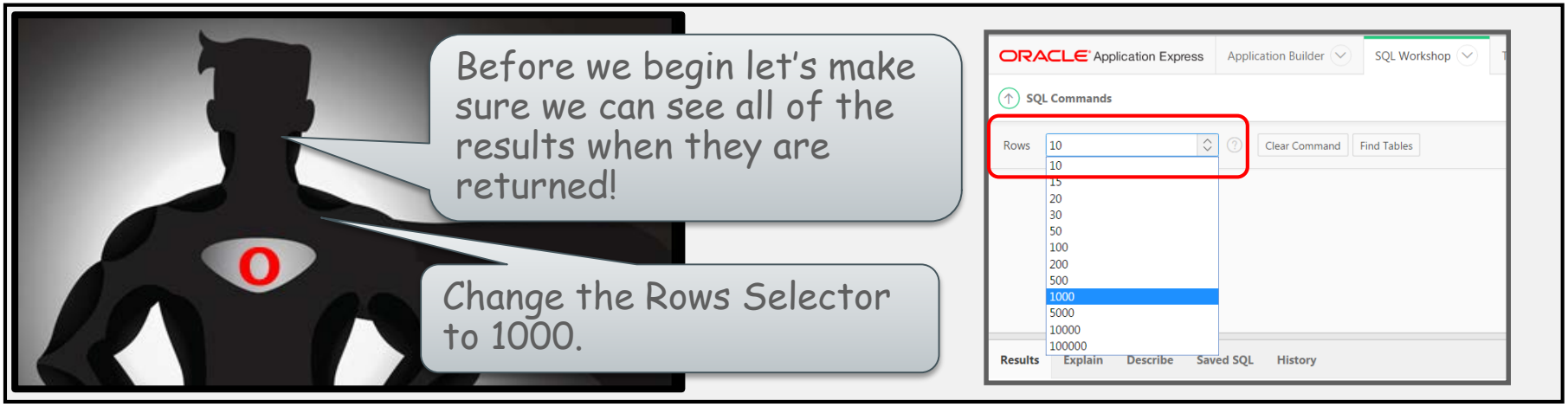


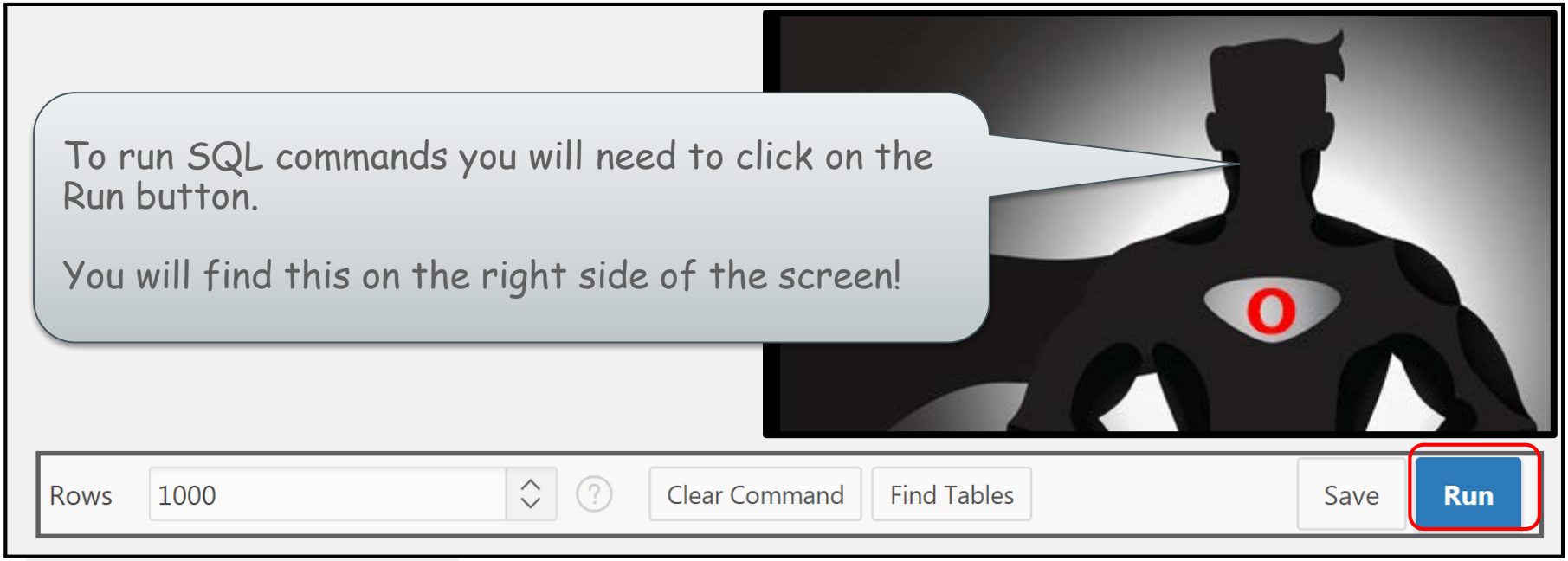
Running a SQL Command



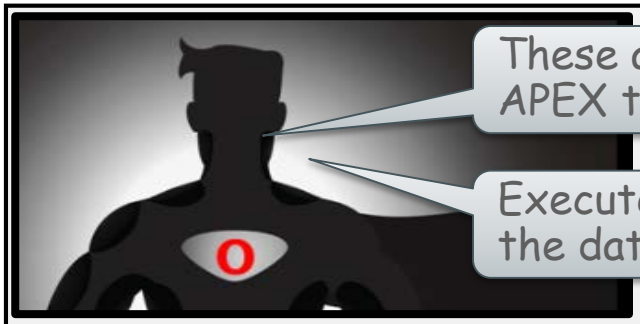


Running a SQL Command





Running SQL Commands



These are the 3 main components of APEX that we will be working with.

Execute them in the given order to return the data you asked for from the table.

2 Click the Run button.

1 Enter the command in the command editor.

ORACLE Application Express

Application Builder | SQL Workshop | Team Development | Packaged Apps

SQL Commands — Schema: WMCRAE_SQL_16

Rows: 1000 | Clear Command | Find Tables | Save | Run

```
SELECT *
FROM suspects;
```

Results | Explain | Describe | Saved SQL | History

SUSPECT_ID	NAME	SEX	AGE	HEIGHT	HAIR_COLOR	EYE_COLOR	FACIAL_HAIR	TATTOOS	GLASSES	SCARS
210	Nasim Jennings	Female	21	Short	Blonde	Blue	No	Yes	No	Yes
211	Mia Greer	Female	78	Short	Black	Brown	Yes	No	No	Yes
212	Byron	Male	18	Short	Black	Green	Yes	No	Yes	Yes

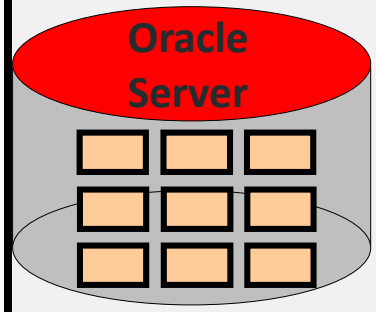
3 View the output on the Results table.

Basic SELECT Statement

The first SQL statement that we will look at is the **SELECT** statement.

We use a **SELECT** statement to get information out of the database.

The **SELECT *** command returns all the rows in a table.



The syntax is:

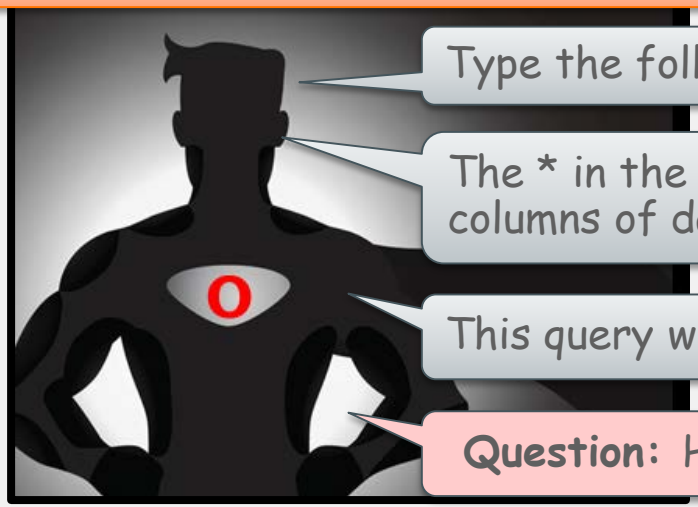
```
SELECT *
FROM <table name>;
```

For example:

```
SELECT *
FROM suspects;
```

Basic SELECT Statement

It's time for you to help SQLman...



Type the following command in the SQL command space.

The * in the SELECT statement means that you will return all columns of data from the table.

This query will return all of the data held in the suspects table!

Question: How many rows were returned?

Rows 1000 Clear Command Find

```
SELECT *
FROM suspects;
```

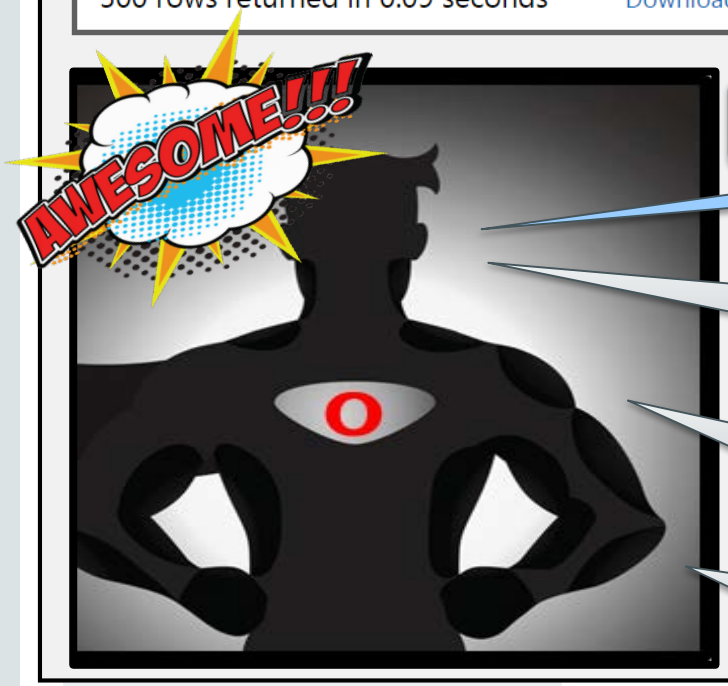
Click the **Run** button to view the results!!

Viewing the results

Results | Explain | Describe | Saved SQL | History

207	Myles Tucker	Male	85	Short	Blonde	Green	No	Yes
208	Rafael Fisher	Female	85	Short	Red	Green	Yes	No
209	Aretha Perkins	Female	55	Medium	Black	Brown	Yes	No

500 rows returned in 0.09 seconds Download



Great Job! We have 500 suspects in our table.

You can see the number of results by scrolling down to the bottom of the result table.

Here it also tells you how long the query took to run.

Now we need to narrow it down to only **two**!!

Correcting errors

When entering SQL commands, it is important to use the correct spelling, otherwise you will get an error message.

For example typing(SEECT instead of SELECT):

```
SEECT *
FROM suspects;
```

Would result in the following error message:

```
ORA-00900: invalid SQL statement
```

To rectify, simply correct the spelling and run again.

Writing SQL Statements

SQL statements are not case-sensitive.

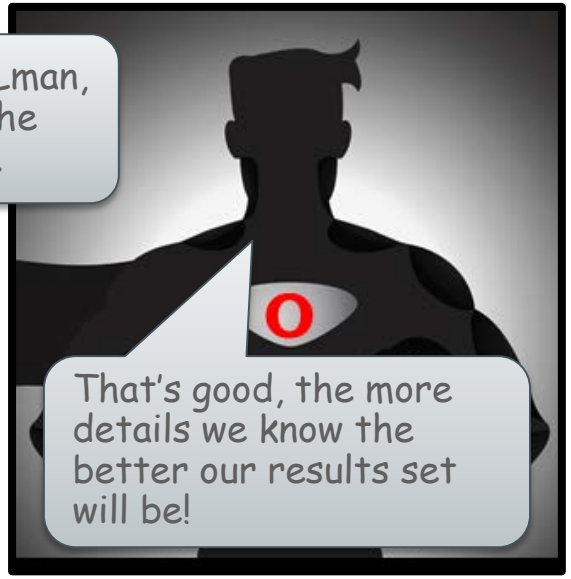
SQL statements can be entered on one or more lines.

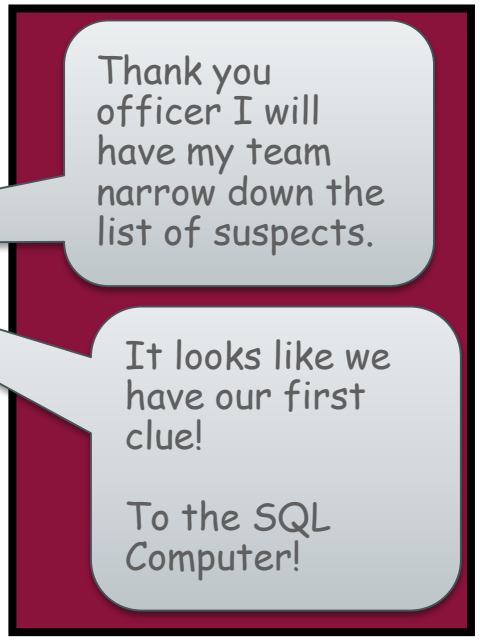
Keywords cannot be abbreviated or split across lines and are typically spelled with uppercase letters.

Indents are used to enhance readability.

It's the hotline from police headquarters...

Ring! Ring!





Selecting Specific Columns

The first time we used a SELECT statement, we used the * to see all columns in our table.

```
SELECT *
FROM   suspects;
```

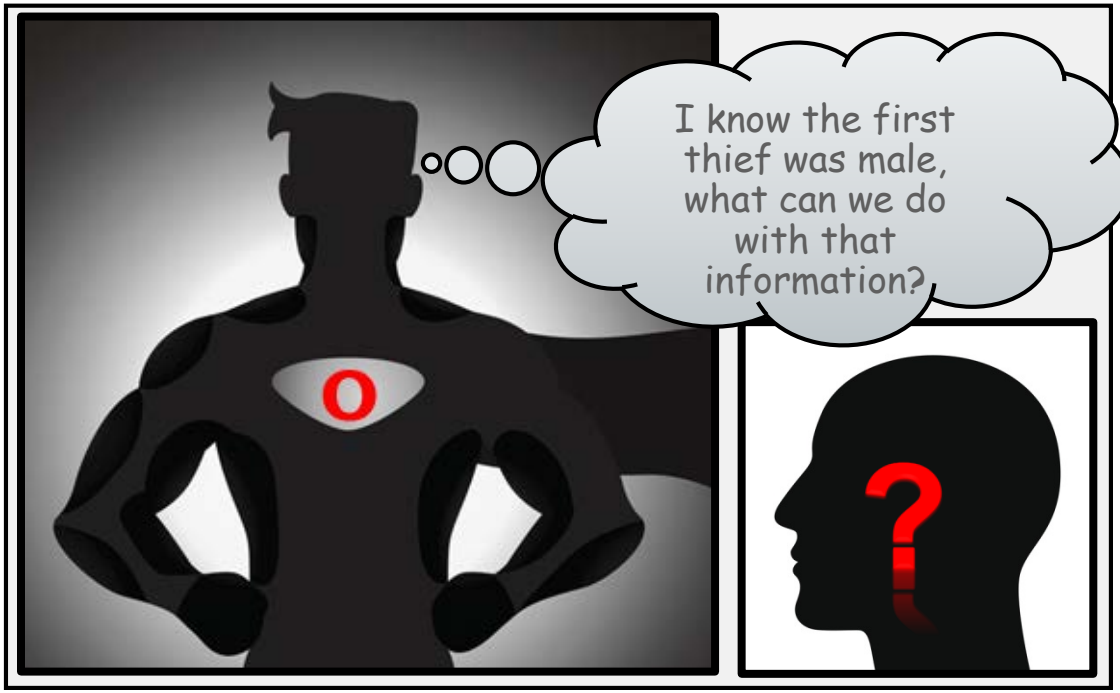
At times we may not require to look at all of the columns in the database table.

This SELECT statement allows us to specify only the columns we need e.g.

```
SELECT name, tattoos
FROM   suspects;
```

This will only show us their name and tattoo information.

Clue #1 The first thief was Male



ACTION :

Write a Select statement that returns all rows but only displays the name and sex columns of the suspects!

Type the following command in the SQL command editor in APEX.

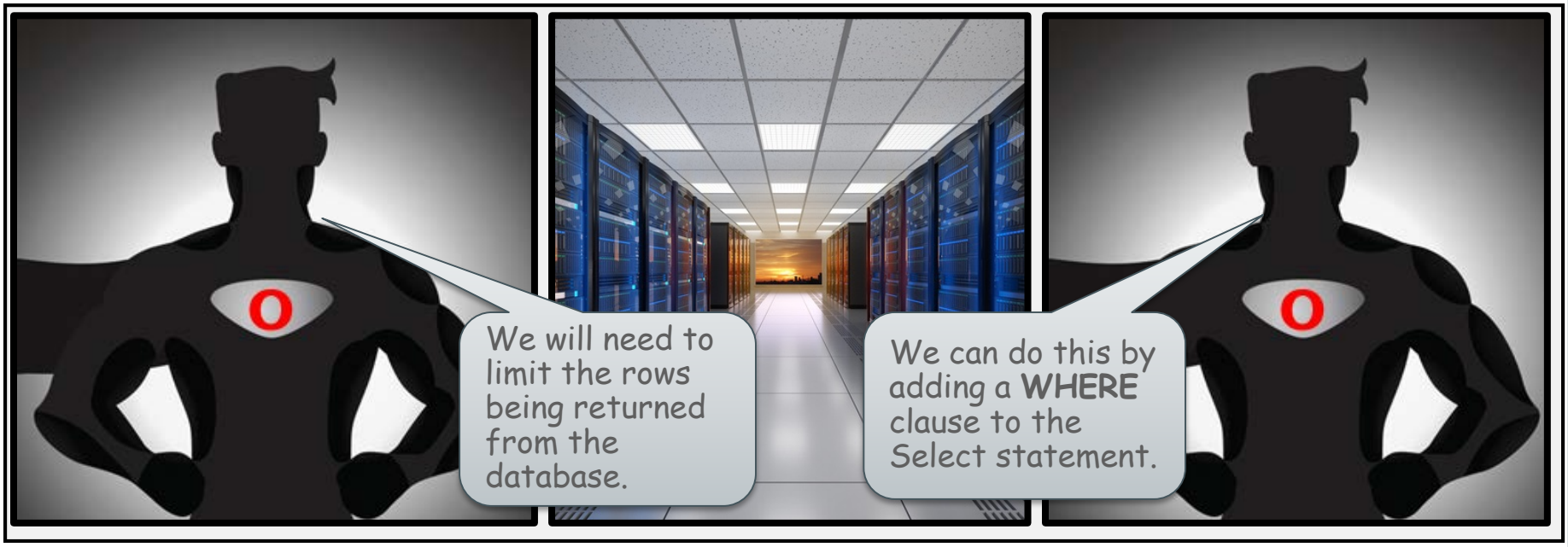
```
SELECT name, sex
FROM   suspects;
```

SQLman evaluates the results...

Results | Explain | Describe | Saved SQL | History

Clinton Fitzgerald	Male
Myles Tucker	Male
Rafael Fisher	Female
Aretha Perkins	Female

500 rows returned in 0.02 seconds Download



Solve It With SQL

Lesson 2
Introduction to SQL

Part 4 – Conditions

Limiting Rows Using a Selection

Assume that you want to display all members in department 90 from an members table. The rows with a value of 90 in the department_id column are the only ones that will be returned.

A row is the information held about a single member in our table!!

We restrict the rows that are returned by using the **WHERE** clause

The **WHERE** clause follows the **FROM** clause.

This query will retrieve and display all information about the members who are in department 90.

```
SELECT *
FROM members
WHERE department_id = 90;
```

MEMBER_ID	FIRST_NAME	LAST_NAME	JOB_ID	DEPARTMENT_ID	SALARY
100	Steven	King	AD_PRES	90	24000
101	Neena	Kochhar	AD_VP	90	17000
102	Lex	De Haan	AD_VP	90	7000

Using the WHERE Clause

It's time for you to have a quick practice...

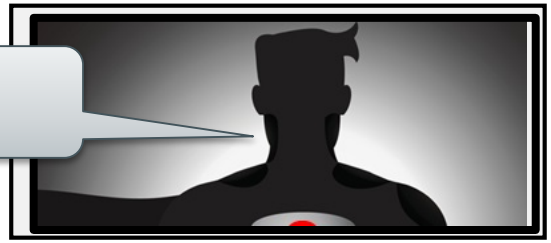
In APEX write the following query in the command editor to see results based on our suspects table.

```
SELECT name, tattoos
FROM   suspects
WHERE  tattoos = 'Yes';
```

NAME	TATTOOS
Orli Parks	Yes
TaShya Witt	Yes
Quemby Ewing	Yes
Eliana Mccall	Yes

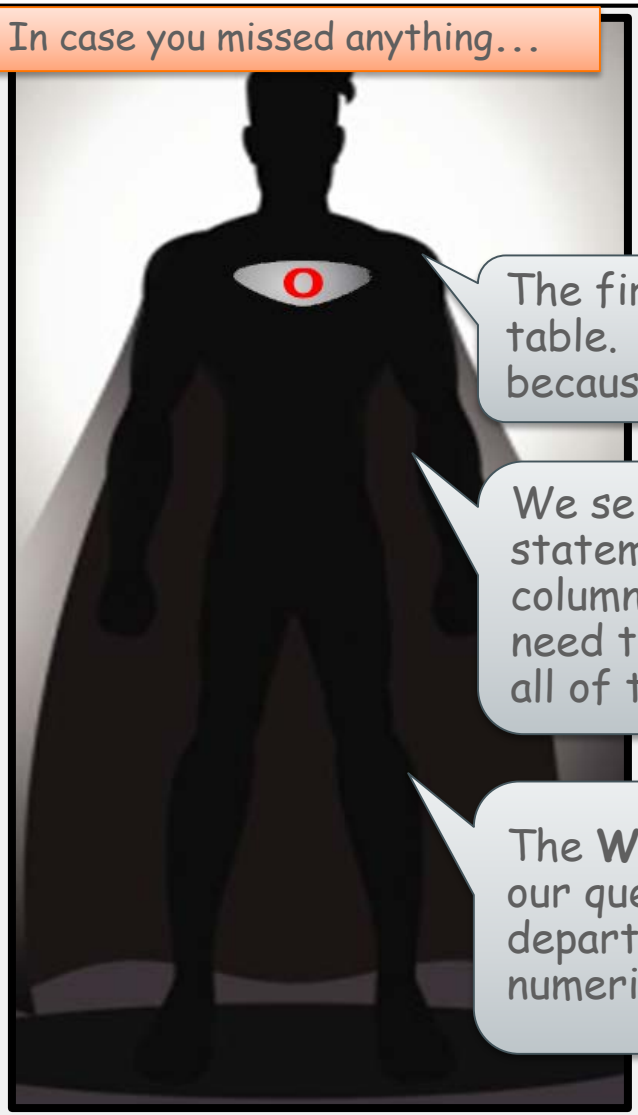
Only the suspects with tattoos are now shown!

There are 250 suspects with tattoos. That's half of the people on our list!!



Quick Recap

In case you missed anything...



```
SELECT *
FROM members
WHERE department_id = 90;
```

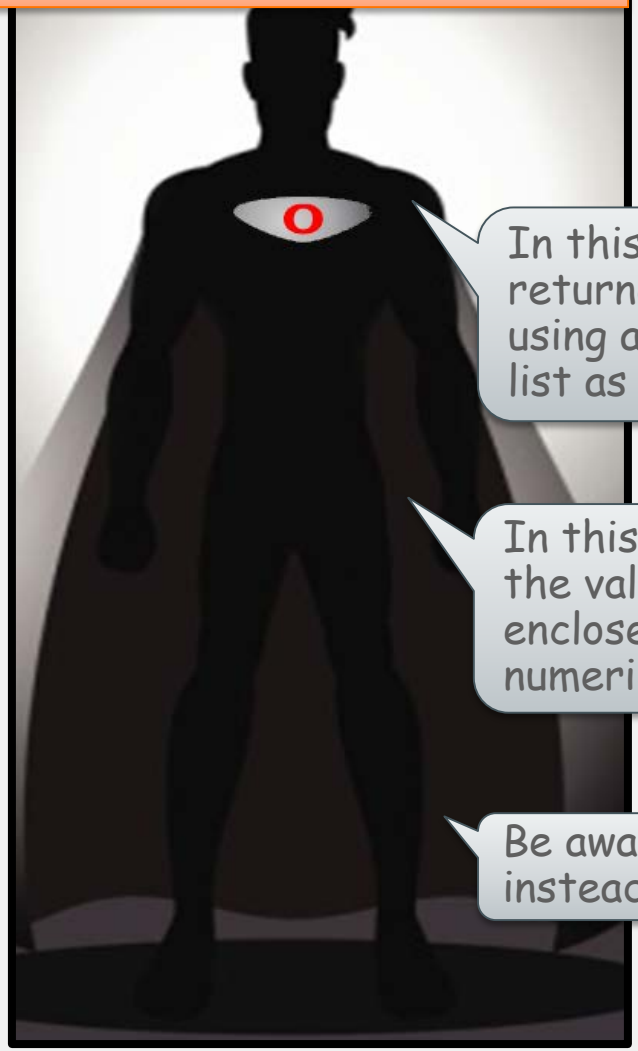
The first query with a **WHERE** clause used the members table. We know which table the information is coming from because we specify it in the **FROM** clause.

We selected all the columns by using the ***** in the **SELECT** statement. The **SELECT** statement is where we specify the columns that we want to be returned by our query. We don't need to know the names of the columns when using the * as all of the columns from the table will be returned.

The **WHERE** clause restricts the number of rows returned by our query. In this query we restricted the rows to only department 90 members. This was done by providing the numeric value 90 for the department_id.

Quick Recap cont..

More about query 2...



```
SELECT name, tattoos
FROM   suspects
WHERE  tattoos = 'Yes';
```

In this query we didn't use the * for all columns to be returned. Instead we named them and separated them by using a comma. Don't put a comma at the end of the column list as this will give you an error!!

In this **WHERE** clause we match text (Yes) in our query. If the value in the WHERE clause is not numeric then we have to enclose it in single quotation marks ' '. We don't do this with numeric values, '90' would give us an error.

Be aware that character values are case-sensitive, using 'yes' instead of 'Yes' would return zero results!!

Quick Recap cont..

Answer the following questions …

```
SELECT name, tattoos
FROM   suspects
WHERE  tattoos = 'Yes';
```

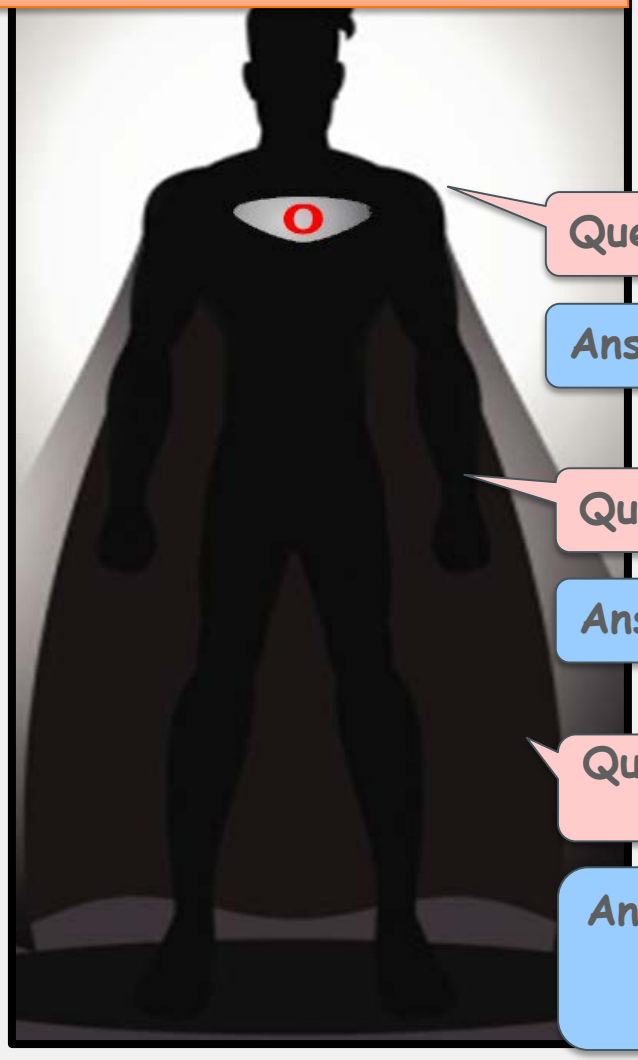
Question 1: What table is used in this query?

Question 2: What columns are returned in this query?

Question 3: How would we return the suspects that don't have tattoos?

Quick Recap cont..

Did your answers match mine?



```
SELECT name, tattoos
FROM   suspects
WHERE  tattoos = 'Yes';
```

Question 1: What table is used in this query?

Answer 1: suspects

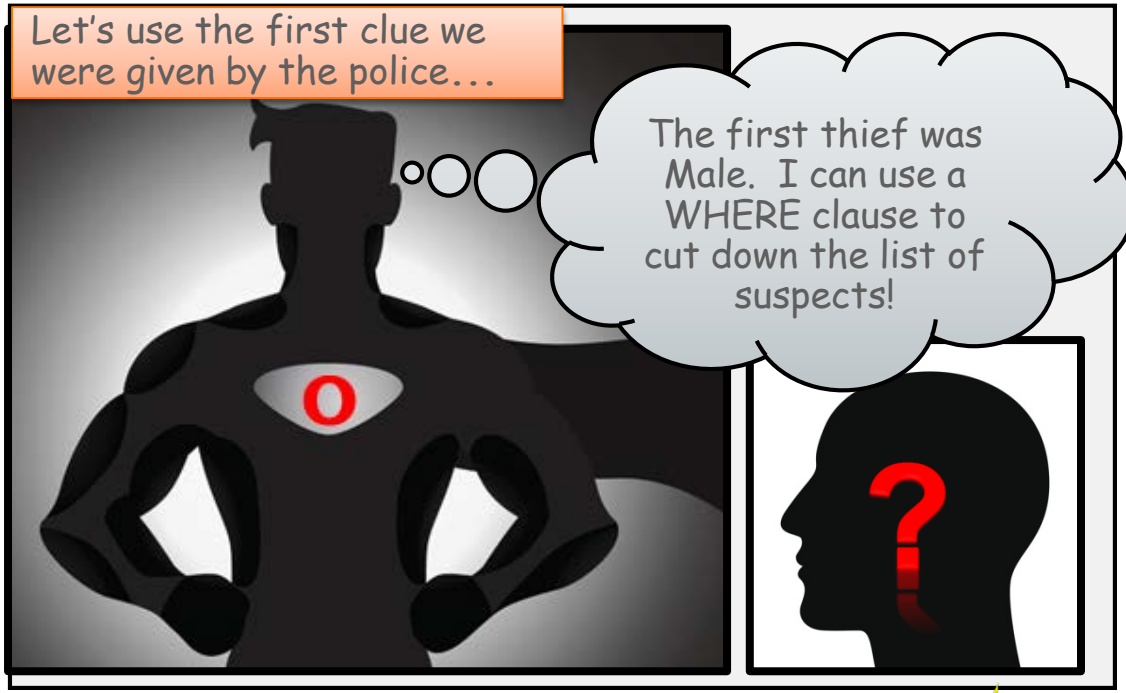
Question 2: What columns are returned in this query?

Answer 2: name and tattoos

Question 3: How would we return the suspects that don't have tattoos?

Answer 3:
```
SELECT name, tattoos
FROM   suspects
WHERE  tattoos = 'No'
```

Clue #1 The first thief was Male



ACTION :

Write a Select statement that returns the **name** and **sex** columns from the **suspects** table of the **Male** suspects!

Type the following command in the command editor in APEX

```
SELECT name, sex
FROM   suspects
WHERE  sex = 'Male';
```

It's the hotline from police headquarters...

Thank you officer I will have my team narrow down the list of suspects even more.

It looks like we now have two clues!

For this we need to learn about Conditions using Logical Operators.

Solve It With SQL

Lesson 2
Introduction to SQL

Part 5 – Conditions using Logical Operators

Conditions Using Logical Operators

A logical operator combines the result of two conditions to produce a single result based on those conditions.

A row is returned from the table only if the overall result of the condition is true.

So far we have specified only one condition in the WHERE clause. You can use several conditions in a single WHERE clause by using the logical operators.

The three logical operators available are:
AND,
OR,
NOT

Operator	Meaning
AND	Returns TRUE if both component conditions are TRUE
OR	Returns TRUE if either component condition is TRUE
NOT	Returns TRUE if the condition is FALSE Returns FALSE if the condition is TRUE

Using the Logical AND Operator

AND requires both component conditions to be true:

```
SELECT member_id, last_name, job_id, salary
FROM   members
WHERE  salary >= 10000
AND    job_id LIKE '%MAN%';
```

This **WHERE** clause will return only members who earn 10000 or over. The **LIKE** operator is used here to search for a specific pattern (MAN) in the text of the job_id column.

I have used the greater than or equal to sign (>=) in the WHERE clause to cover a range of salaries

MEMBER_ID	LAST_NAME	JOB_ID	SALARY
149	Zlotkey	SA_MAN	10500
201	Hartstein	MK_MAN	13000

Remember character searches are case-sensitive. If 'MAN' is not stored in uppercase characters in the table then no rows will be returned!!

Using the Logical NOT Operator

NOT will return TRUE if the condition is FALSE or return FALSE if the condition is TRUE.

```
SELECT member_id, last_name, job_id, salary
FROM   members
WHERE  salary >= 10000
AND    job_id NOT LIKE '%MAN%';
```

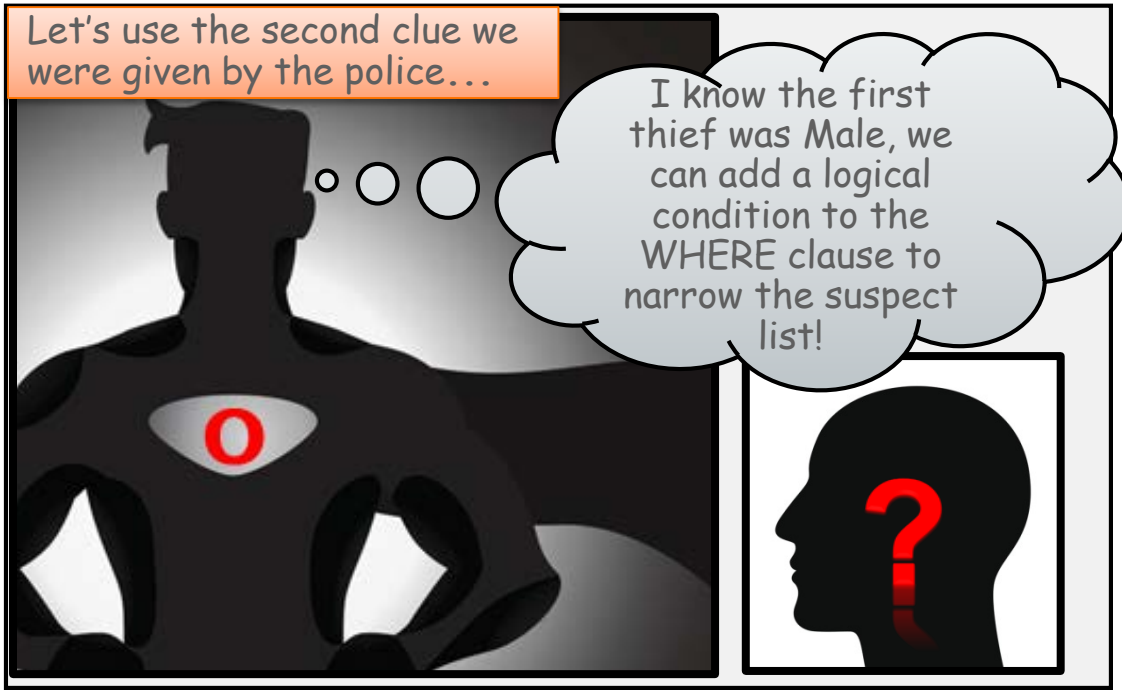
This would return employees who earn 10000 or over and do not have MAN somewhere in their job title.

The results would look like this.

MEMBER_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
205	Higgins	AC_MGR	12000
174	Abel	SA_REP	11000

Remember using **NOT** will return the opposite values.
A condition that would have returned TRUE will now return FALSE!!

Clue #2 The first thief also had scars

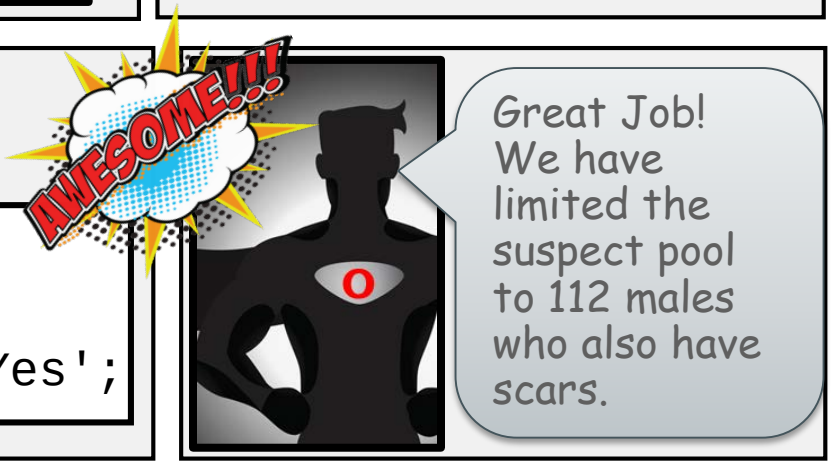


ACTION :

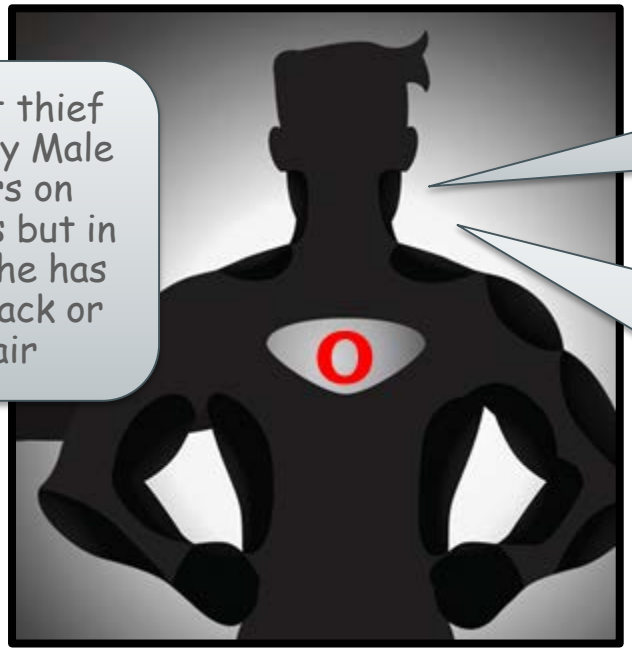
Write a Select statement that returns the **name**, **sex** and **scars** columns from the suspects table of the **Male** suspects who have also have **scars**!

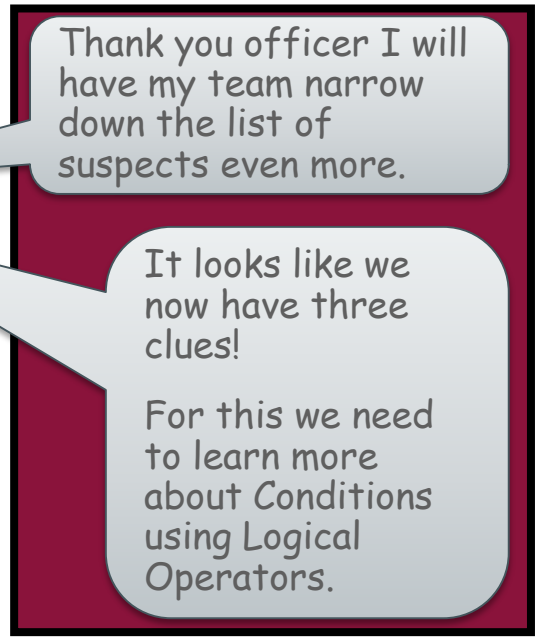
Type the following command in the command editor in APEX

```
SELECT name, sex, scars
FROM   suspects
WHERE sex = 'Male' AND scars = 'Yes';
```



It's the hotline from police headquarters...





Using the Logical OR Operator

OR requires one of the component conditions to be true:

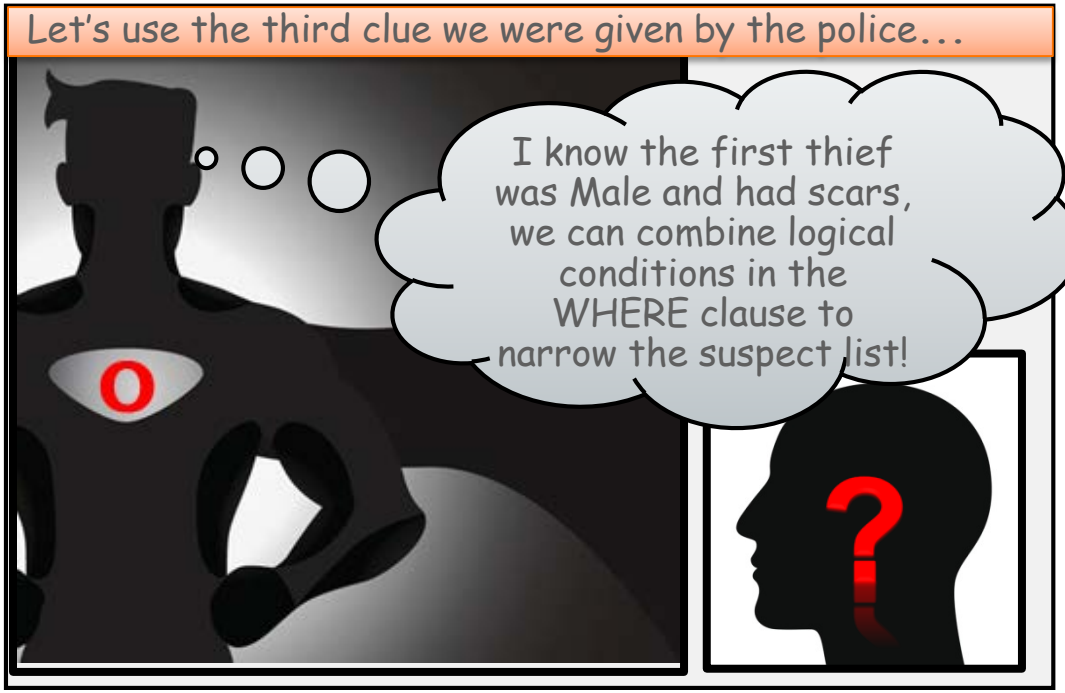
```
SELECT member_id, last_name, job_id, salary
FROM   members
WHERE  salary >= 10000
OR     job_id LIKE '%MAN%';
```

This would return employees who earn 10000 or over **OR** have MAN somewhere in their job title.

MEMBER_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
205	Higgins	AC_MGR	12000
149	Zlotkey	SA_MAN	10500
174	Abel	SA_REP	11000
124	Mourgos	ST_MAN	5800
201	Hartstein	MK_MAN	13000

Remember using **AND** means both have to be true. Using **OR** means that only one of the components has to be true!!

Clue #3 Thief has either Black or Brown hair.

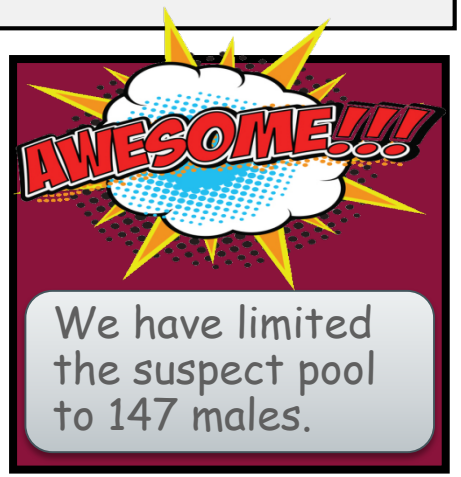


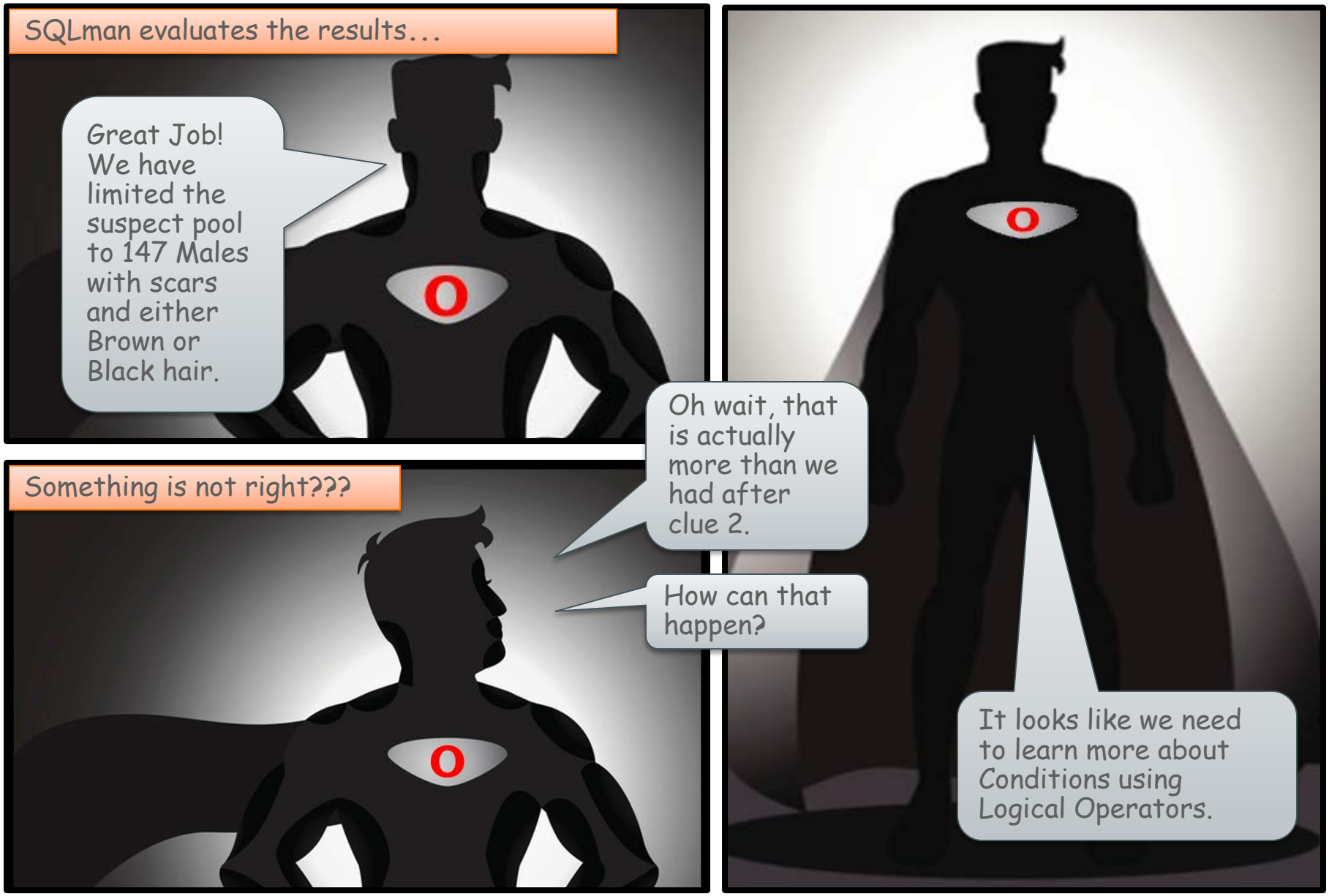
ACTION :

Write a Select statement that returns the **name**, **sex, scars** and **hair color** columns from the **suspects** table of **Male** suspects who have **scars** and either **Brown** or **Black** Hair!

Type the following command in the command editor in APEX

```
SELECT name, sex, scars, hair_color
FROM   suspects
WHERE  sex = 'Male' AND scars = 'Yes' AND
hair_color = 'Brown' OR hair_color = 'Black';
```





Solve It With SQL

Lesson 2
Introduction to SQL

Part 6 – Rules of Precedence

Rules of Precedence

The rules of precedence determine the order in which expressions are evaluated and calculated.

The table in this slide lists the default order of precedence

To override the default order we can enclose expressions in parentheses so that they are calculated first.

Precedence	Operator
1	Arithmetic operators
2	Concatenation operator
3	Comparison conditions
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Not equal to
7	NOT logical operator
8	AND logical operator
9	OR logical operator

Rules of Precedence

In Clue #3 the **AND** was evaluated before the OR giving us this SQL statement.

```
SELECT name, sex, scars, hair_color
FROM suspects
WHERE sex = 'Male' AND scars = 'Yes' AND
hair_color = 'Brown';
```

For Brown hair **26 Rows** were returned!

The **OR** was then evaluated giving us this SQL statement.

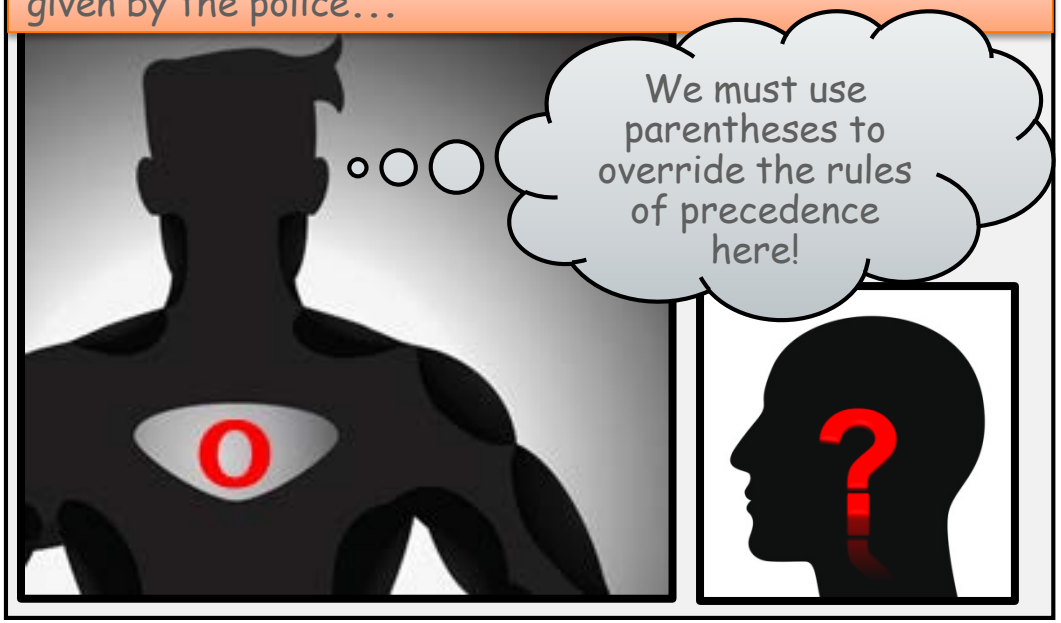
```
SELECT name, sex, scars, hair_color
FROM suspects
WHERE hair_color = 'Black';
```

For Black hair **121 Rows** were returned!

Both sets of results were then combined and returned to us. That is why we had a total of 147 results!!

Clue #3 Thief has either Black or Brown hair.

Let's try that query again using the third clue we were given by the police...



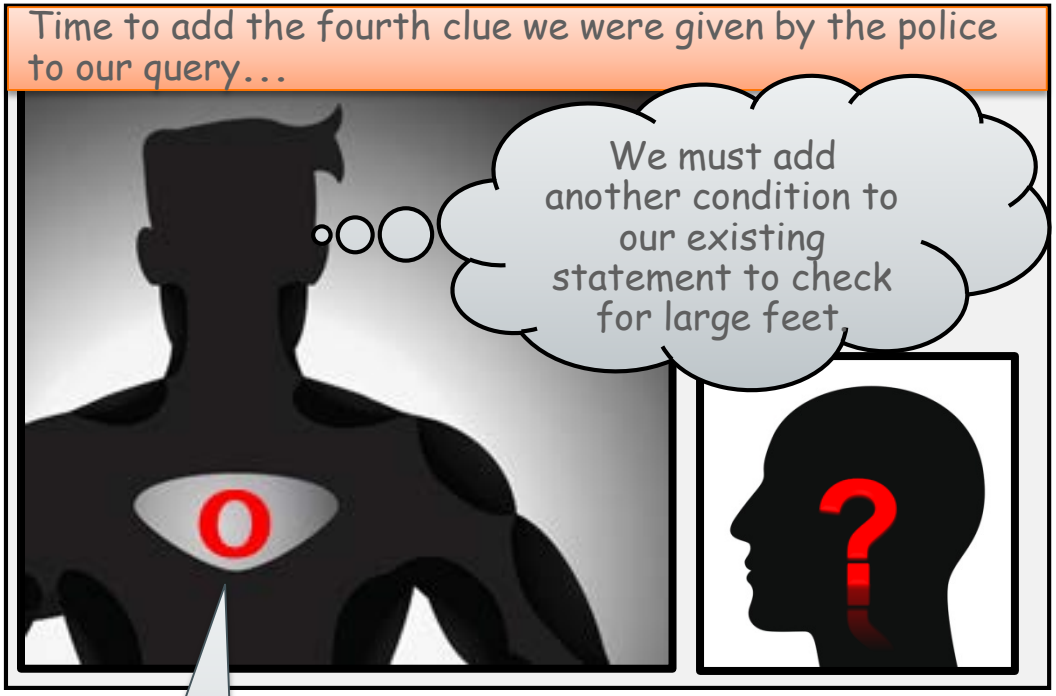
ACTION :

Write a Select statement That returns the **name**, **sex, scars** and **hair color** columns from the **suspects** table of **Male** suspects who have **scars** and either **Brown** or **Black** Hair!

Type the following command in the command editor in APEX

```
SELECT name, sex, scars, hair_color
FROM   suspects
WHERE  sex = 'Male' AND scars = 'Yes' AND
(hair_color = 'Brown' OR hair_color = 'Black');
```

Clue #4 The thief also has Large feet..



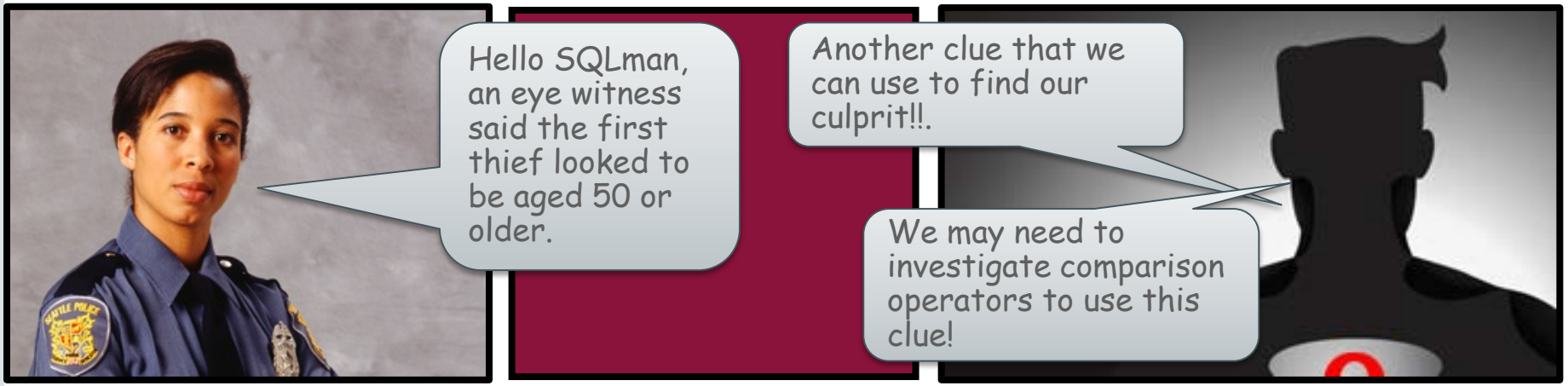
ACTION :

Write a Select statement that returns the **name**, **sex**, **scars, hair color** and **feet size** columns from the **suspects** table of **Male** suspects who have **scars** and either **Brown** or **Black** Hair and **Large** feet!

Type this command in the command editor in APEX.

```
SELECT name, sex, scars, hair_color, feet_size
FROM   suspects
WHERE sex = 'Male' AND scars = 'Yes' AND
(hair_color = 'Brown' OR hair_color = 'Black') AND
feet_size = 'Large';
```

SQLman gets his wish...



Solve It With SQL

Lesson 2
Introduction to SQL

Part 7 – Comparison Operators

Comparison Operators

Comparison operators are used in conditions that compare one expression with another value or expression.

You are already familiar with some of these operators as we have used = and >= already!

They are used in the WHERE clause.

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to
BETWEEN…AND…	Between two values (inclusive)
IN (set)	Match any of a list of values
LIKE	Match a character pattern
IS NULL	Is a null value

Using Comparison Operators

With these skills we won't be long in catching the thief now!!

In this example, the SELECT statement retrieves the last name and salary from the MEMBERS table for any member whose salary is less than or equal to the explicit value 3,000.

```
SELECT last_name, salary
FROM   members
WHERE  salary <= 3000;
```

LAST_NAME	SALARY
Matos	2600
Vargas	2500

You can use comparison and logical operators together!

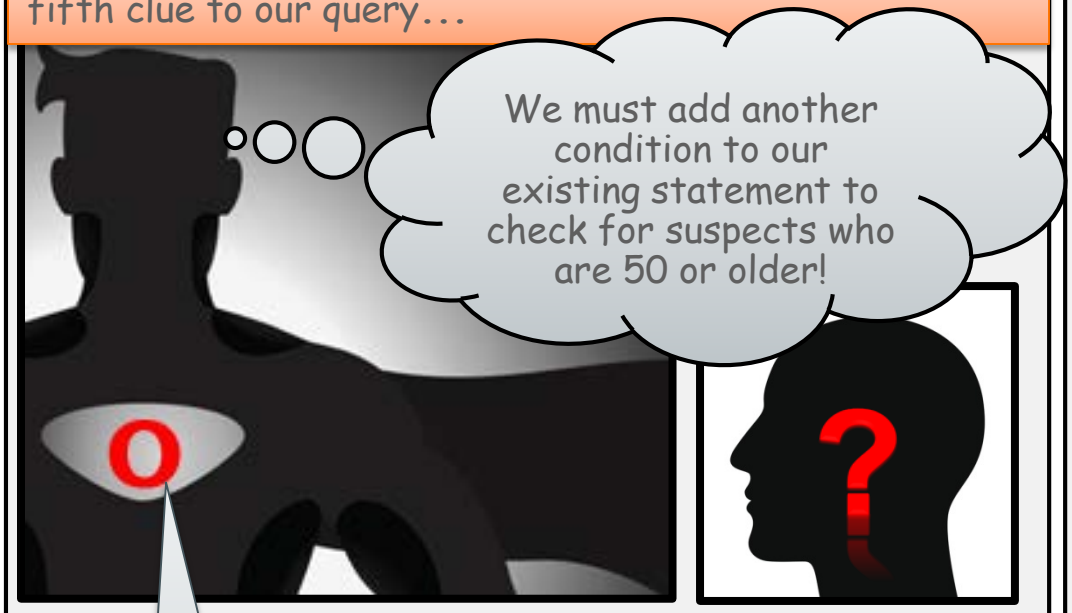
```
SELECT last_name, salary
FROM   members
WHERE  salary <= 3000 AND SALARY > 2500;
```

LAST_NAME	SALARY
Matos	2600

To see how we put these skills into practice go to the next slide. . .

Clue #5 The first thief looked aged 50 or older

Let's see what difference it makes when we add the fifth clue to our query...



ACTION :

Write a Select statement that returns the **name**, **sex**, **scars, hair color, feet size** and **age** columns from the **suspects** table of **Male** suspects who have **scars** and either **Brown** or **Black** Hair and **Large** feet and are either **50** years old or older!

Type this command in the command editor in APEX.

```
SELECT name, sex, scars, hair_color, feet_size, age
FROM   suspects
WHERE sex = 'Male' AND scars = 'Yes' AND
(hair_color = 'Brown' OR hair_color = 'Black') AND
feet_size = 'Large' AND age >=50;
```

Range Conditions: BETWEEN Operator

Use the **BETWEEN** operator to display rows based on a range of values within a lower and an upper limit.

We have used something similar before using **AND** or **OR** but with the **BETWEEN** operator we don't need to rewrite the field name!

```
SELECT last_name, salary
FROM   members
WHERE  salary BETWEEN 2500 AND 3500 ;
```

Lower limit

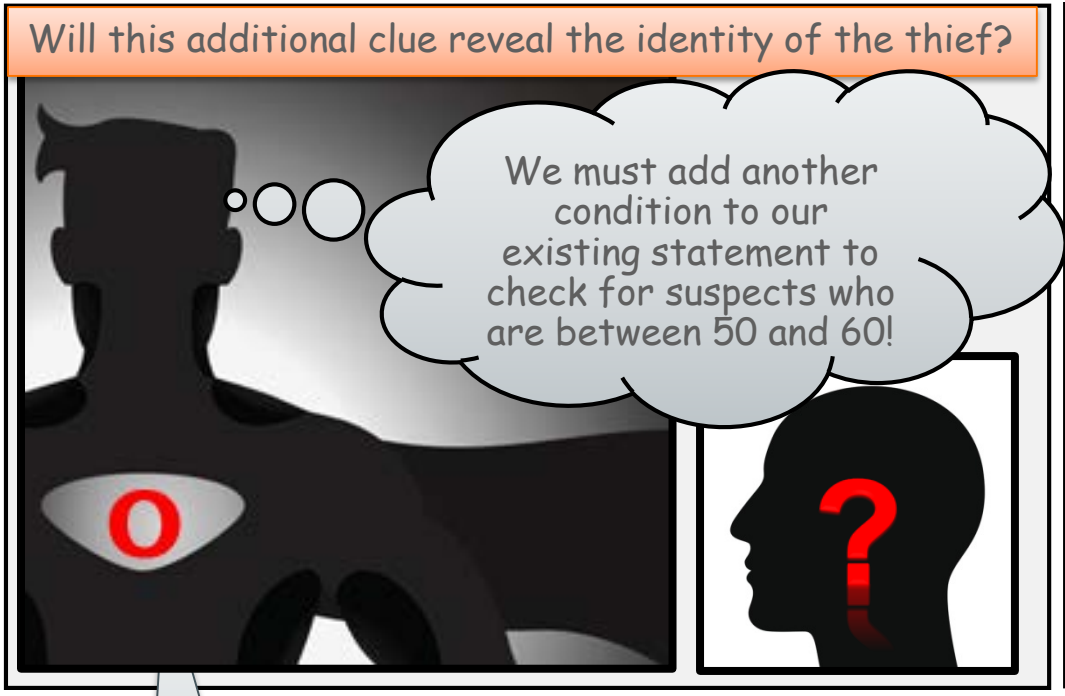
Upper limit

The results would look like this:

LAST_NAME	SALARY
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500

It's important to always place the lowest value first in the BETWEEN clause or your query will return zero results!!

Clue #6 The thief looked between 50 and 60



ACTION :

Write a Select statement that returns the **name**, **sex**, **scars, hair color, feet size** and **age** columns from the **suspects** table of **Male** suspects who have **scars** and either **Brown** or **Black** Hair and **Large** feet and are between **50** and **60** years old!

Type this command in the command editor in APEX.

```
SELECT name, sex, scars, hair_color, feet_size, age
FROM   suspects
WHERE sex = 'Male' AND scars = 'Yes' AND
(hair_color = 'Brown' OR hair_color = 'Black') AND
feet_size = 'Large' AND age BETWEEN 50 AND 60;
```

Clue #7 The first thief wore glasses

Will this additional clue reveal the identity of the thief?

ACTION :

Write a Select statement that returns the **name**, **sex**, **scars, hair color, feet size, age** and **glasses** columns from the **suspects** table of **Male** suspects who have **scars** and either **Brown** or **Black** Hair and **Large** feet and are between **50** and **60** years old and wears **glasses**!

```
SELECT name, sex, scars, hair_color, feet_size, age, glasses
FROM   suspects
WHERE sex = 'Male' AND scars = 'Yes' AND
(hair_color = 'Brown' OR hair_color = 'Black') AND
feet_size = 'Large' AND age BETWEEN 50 AND 60 AND
glasses = 'Yes';
```

To catch a thief!!

Your search criteria returned all of the details to confirm that this was the first thief that we were looking for!

Once we knew some details about him he never stood a chance!!

You can't hide from a SQL database!!!

NAME	SEX	SCARS	HAIR_COLOR	FEET_SIZE	AGE	GLASSES
Kasper Good	Male	Yes	Black	Large	55	Yes

1 rows returned in 0.03 seconds Download

Before we tell the Police I just want to show you one more thing about the power of SQL.

Viewing Specific Columns

Throughout our search for the thief we have displayed the results for each column that we have included in our search criteria.

```
SELECT name, sex, scars, etc..
```

We don't need to view everything in our results.

We can project only the columns we want to see in our results, in this case the **name**.

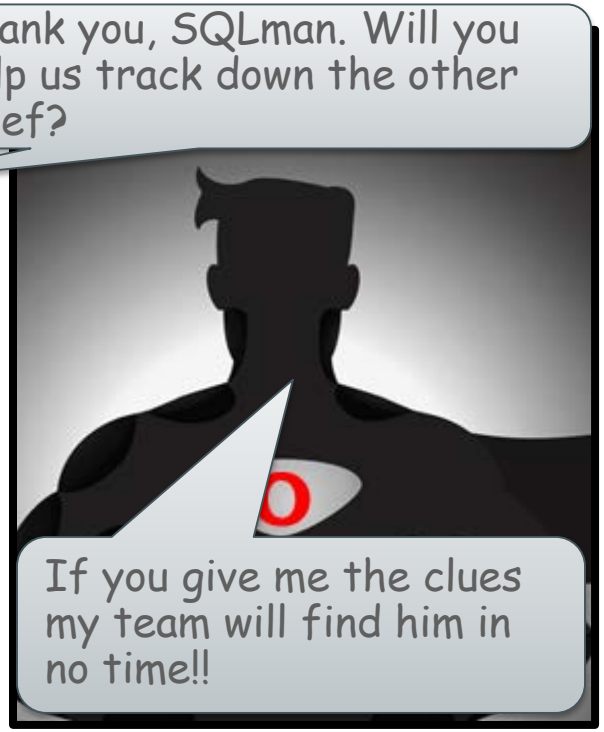
Run this query to only see the name of the thief returned.

```
SELECT name
FROM   suspects
WHERE  sex = 'Male' AND scars = 'Yes' AND
 (hair_color = 'Brown' OR hair_color = 'Black') AND
  feet_size = 'Large' AND age BETWEEN 50 AND 60 AND
  glasses = 'Yes';
```

Kasper Good, you will be going to prison for bank robbery!

We have our mystery hero, a team of coders and SQL to thank for it!

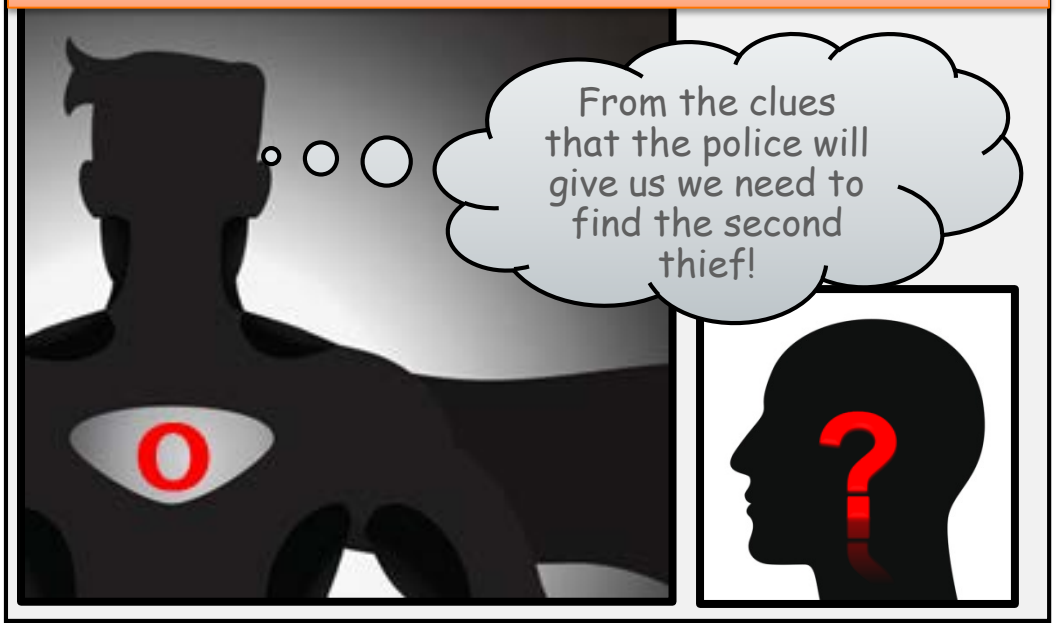
Thank you, SQLman. Will you help us track down the other thief?



If you give me the clues my team will find him in no time!!

Challenge! Find the Second Thief

Time to put what you have learned into practice . . .



ACTION :

Using each clue provided build the SQL statement clue by clue to find the second thief.

After each clue has been added to the query record the number of results returned when you run it.

As I give you each clue I would like to know how many suspects have been returned with that information.

It's amazing how your system works I would love to learn more!

Challenge! Find the Second Thief part 2

Let's catch another thief. .

Every time you get a clue from the police officer build the query in APEX and then run it.

Write down the number of results you get.

When you get down to one then you have your suspect.

I would like you to try this on your own. You can check your results with mine before you pass over the suspect's details to the police.

Okay are you ready for Clue #1?

The suspect has Red hair!

For each clue think about the information you want to display (project) in your result set (name, hair color , etc..).

All the information will come from the suspects table but what columns will you find it in?

Challenge! Find the Second Thief part 3

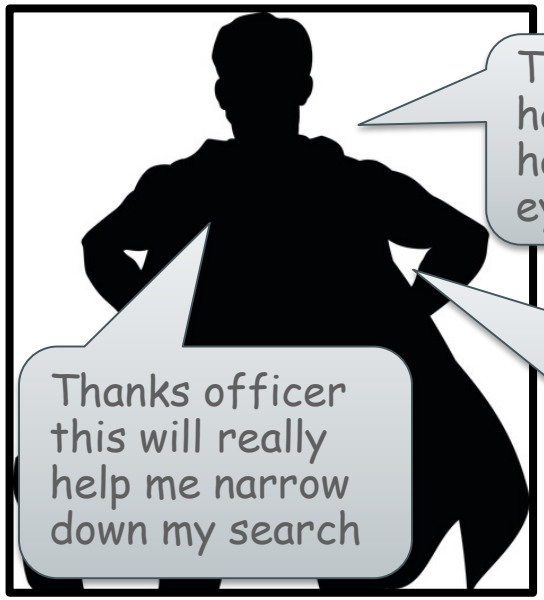
SQLman has faith in you. .

After you have ran each clue write down the following information:
Clue Number :
Number of Results :
Condition Used :

This will help you track your progress

Okay are you ready for Clue #2?

The suspect also has Blue eyes!



The suspect has both red hair AND blue eyes.

There can't be that many suspects with both of those!

Thanks officer this will really help me narrow down my search

Does that mean we have our suspect already?

With a few more clues we will.

Challenge! Find the Second Thief part 4

Okay are you ready for Clue #3?

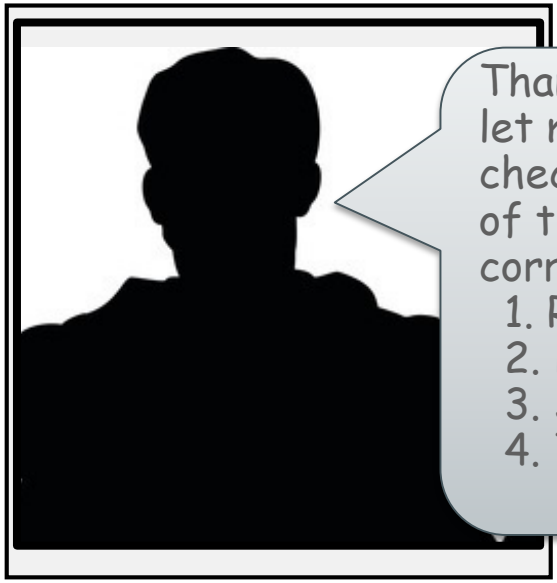
The suspect has Small feet!

Thanks officer, let me design and run the updated query

Once I record the results I'll be ready for the next clue!

I have another clue for you. This makes it Clue #4?

The suspect has a tattoo!



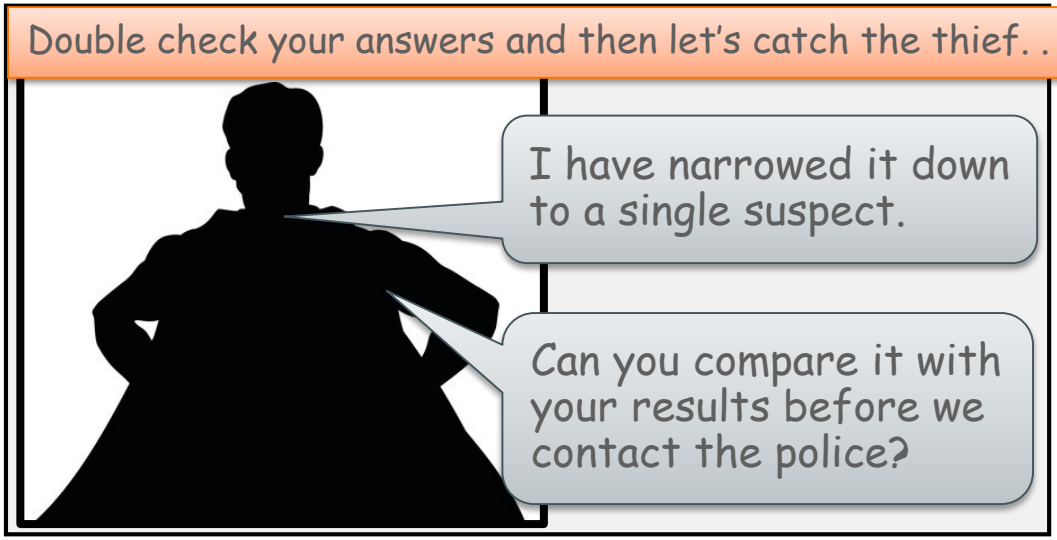
Thanks officer, let me just check I have all of the facts correct so far:

1. Red hair
2. Blue eyes
3. Small feet
4. Tattoo

Challenge! Find the Second Thief part 5

Challenge! Find the Second Thief part 6

Double check your answers and then let's catch the thief. . .



Clue #1 : The suspect has Red hair	SELECT name, hair_color FROM suspects WHERE hair_color = 'Red';	Number of Results: 118
Clue #2 : The suspect has Blue eyes!	SELECT name, hair_color, eye_color FROM suspects WHERE hair_color = 'Red' AND eye_color = 'Blue';	Number of Results: 43

Challenge! Find the Second Thief part 6 cont.

SQLman is checking that your results match his . . .

Clue #3 :
The suspect has Small feet

Number of Results: 9

```
SELECT name, hair_color, eye_color, feet_size
FROM suspects
WHERE hair_color  = 'Red'
AND   eye_color   = 'Blue'
AND   feet_size   = 'Small' ;
```

Clue #4 :
The suspect has a tattoo!

Number of Results: 5

```
SELECT name, hair_color, eye_color, feet_size, tattoos
FROM suspects
WHERE hair_color  = 'Red'
AND   eye_color   = 'Blue'
AND   feet_size   = 'Small'
AND   tattoos     = 'Yes';
```

Challenge! Find the Second Thief part 6 cont.

Check the final clue and compare results. . .

Clue #5 :
The suspect is aged between 35 and 40

Number of Results: 1

```
SELECT name, hair_color, eye_color, feet_size, tattoos, age
FROM suspects
WHERE hair_color  = 'Red'
AND   eye_color   = 'Blue'
AND   feet_size   = 'Small'
AND   tattoos     = 'Yes'
AND   age BETWEEN 35 AND 40;
```

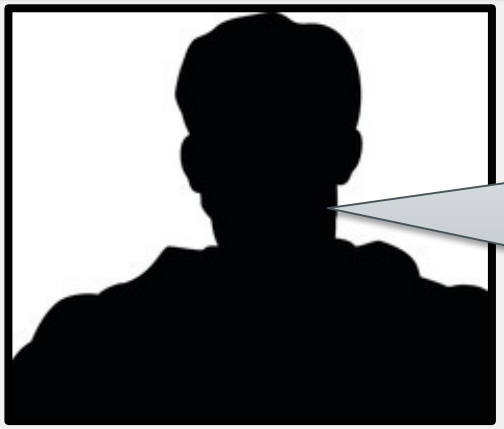
We have narrowed it down to a single suspect.

And that suspect's name is..

Hilary Mayer

Challenge! Second Thief: The final chapter

One final piece of the puzzle is required. . .



We know the suspects name and the information from the clues but are the police looking for a man or a woman?

It's easy to find out that information.

A quick change to our final query and the police will have all the information they need!

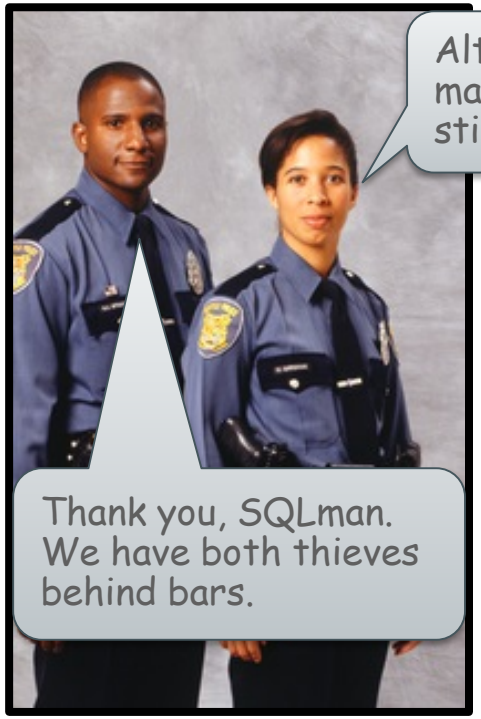
Only SELECT the information we need!!!

```
SELECT name, sex
FROM suspects
WHERE hair_color  = 'Red'
AND   eye_color   = 'Blue'
AND   feet_size   = 'Small'
AND   tattoos     = 'Yes'
AND   age BETWEEN 35 AND 40;
```

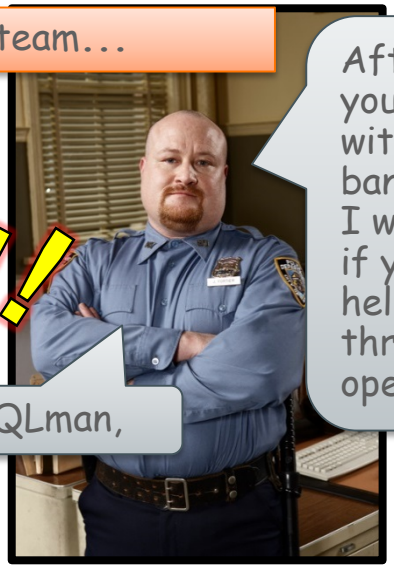
The search criteria returned the details we needed to confirm that this was the second thief that we were looking for!

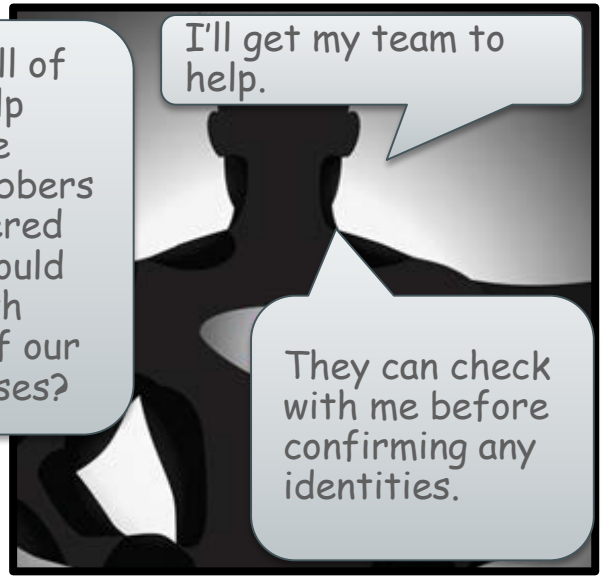
Just like his accomplice, once we knew some details about him he never stood a chance!!

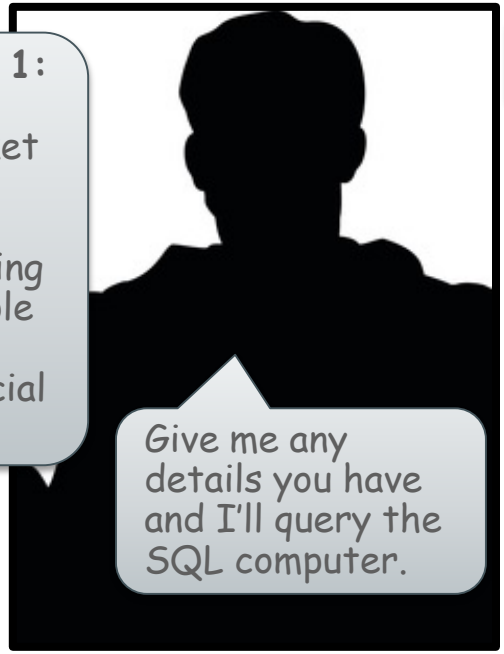
Remember you can't hide from a SQL database!!!

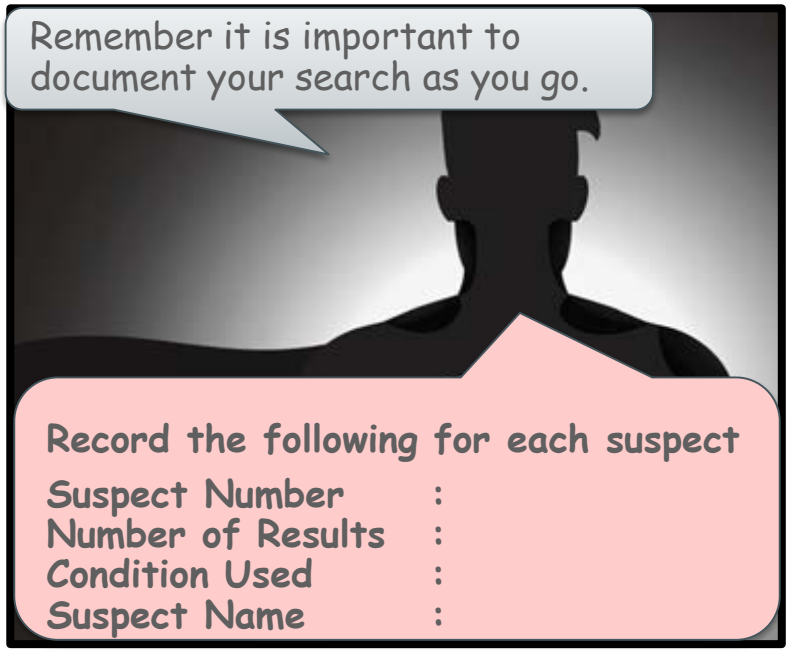


Looks like there is no rest for the SQL team...









The police force have more requests...

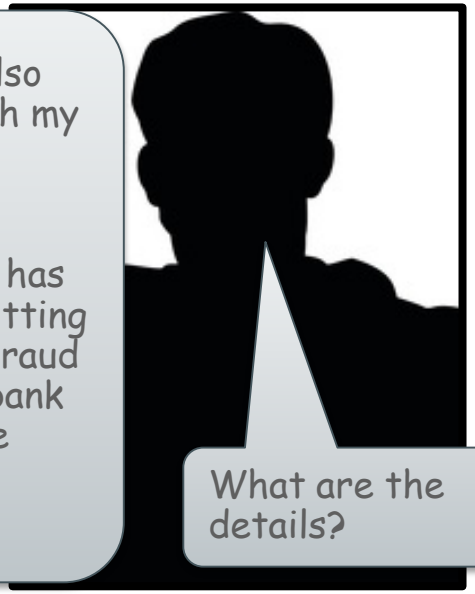
These are the details we have:

Hair color: Blonde
Eye color: Blue
Glasses: No
Feet size: Small
Tattoos: Yes
Age: older than 50

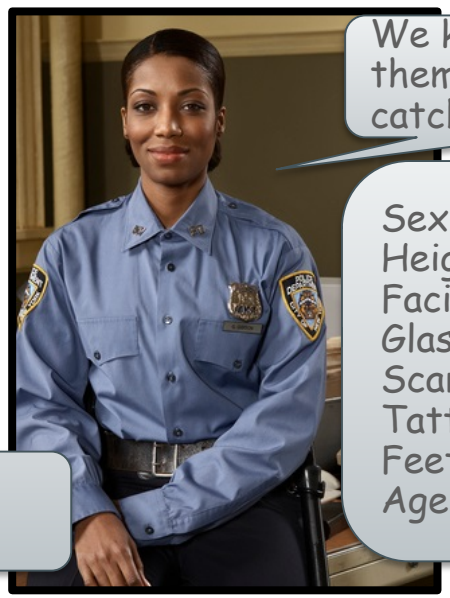
I'll create a query from all that information and hopefully a name will come up!

What are the details for the other suspects

Could you also help me with my case?
Suspect 2: There is a person who has been committing bank card fraud by cloning bank cards in the financial district.



What are the details?



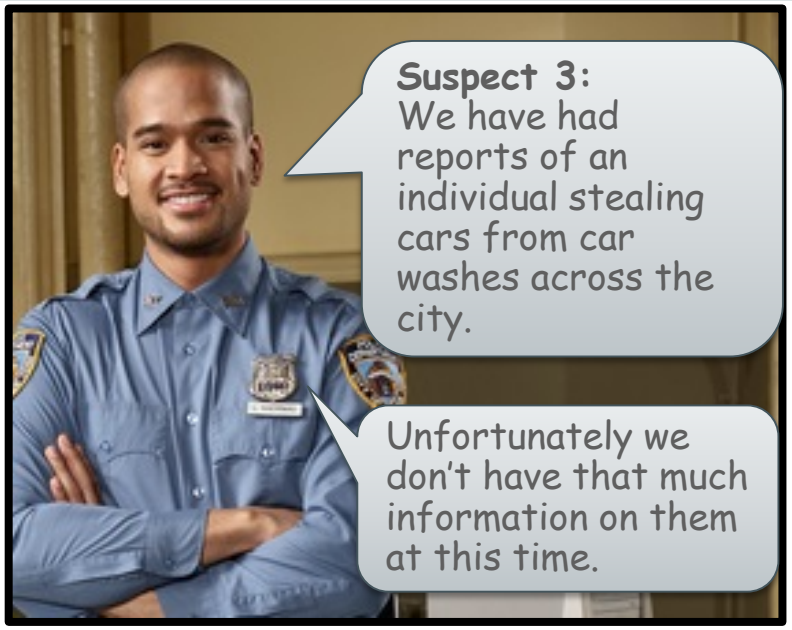
We know lots about them we just can't catch them.

Sex: Female
Height: Tall
Facial hair: Yes
Glasses: No
Scars: Yes
Tattoos: Yes
Feet size: Large
Age: less than 30

The police force have one more request...

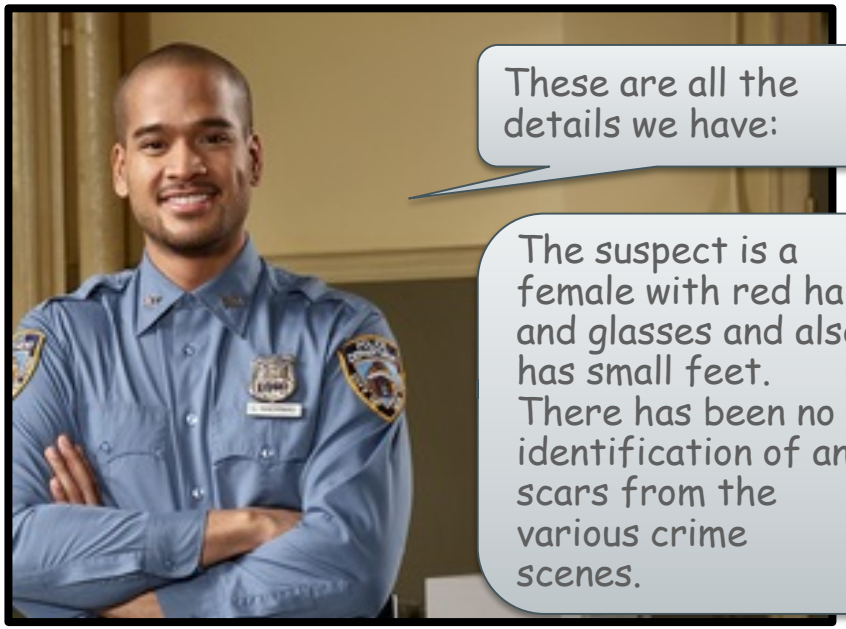
I'll create a query from all that information and hopefully a name will come up!

What are the details for the final suspect?



Suspect 3: We have had reports of an individual stealing cars from car washes across the city.

Unfortunately we don't have that much information on them at this time.



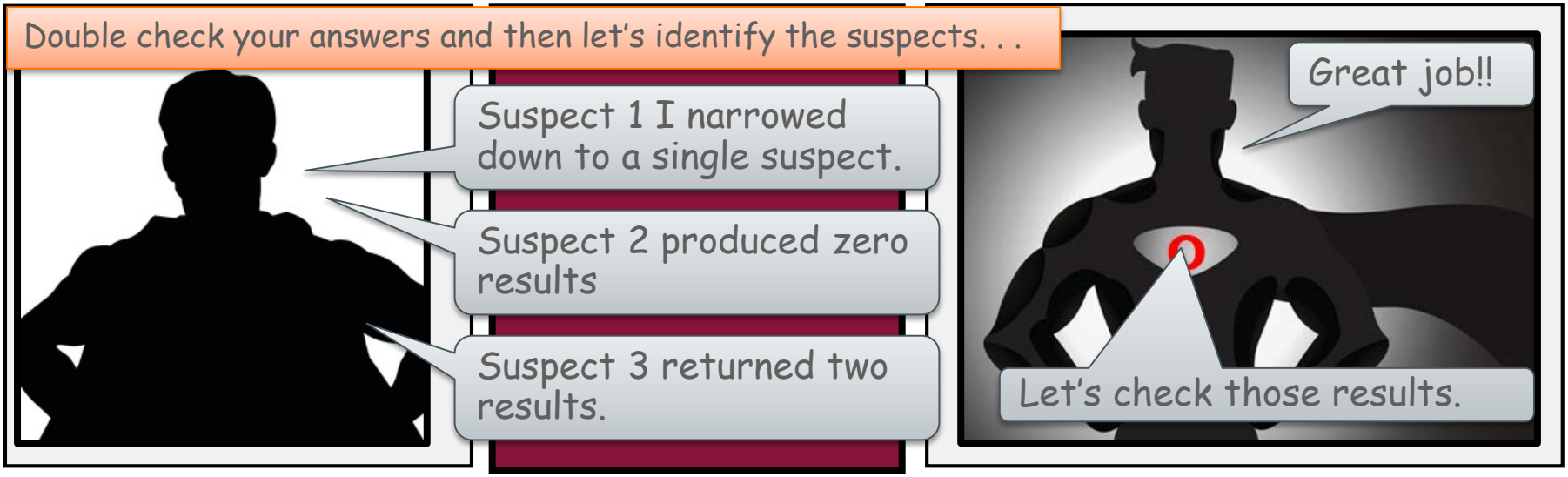
These are all the details we have:

The suspect is a female with red hair and glasses and also has small feet. There has been no identification of any scars from the various crime scenes.

Okay I'll do what I can.

See if you can identify these suspects before you compare results with SQLman on the next slide...

Challenge! Did you identify the suspect?



Suspect #1 :
Suspect's details:
Hair color: Blonde
Eye color: Blue
Glasses: No
Feet size: Small
Tattoos: Yes
Age: older than 50

```
SELECT name, sex
FROM suspects
WHERE hair_color  = 'Blonde'
AND   eye_color   = 'Blue'
AND   glasses     = 'No'
AND   feet_size   = 'Small'
AND   tattoos     = 'Yes'
AND   age > 50;
```

Number of Results: 1

Suspect details:
Brian Russo Male

Challenge! Did you identify the suspect?

Suspect #2 :
Suspect's details:
Sex: Female
Height: Tall
Facial hair: Yes
Glasses: No
Scars: Yes
Tattoos: Yes
Feet size: Large
Age: less than 30

```
SELECT name
FROM suspects
WHERE sex        = 'Female'
AND   height     = 'Tall'
AND   facial_hair = 'Yes'
AND   glasses    = 'No'
AND   scars      = 'Yes'
AND   tattoos    = 'Yes'
AND   feet_size  = 'Large'
AND   age < 30;
```

Number of Results: 0

Suspect details:

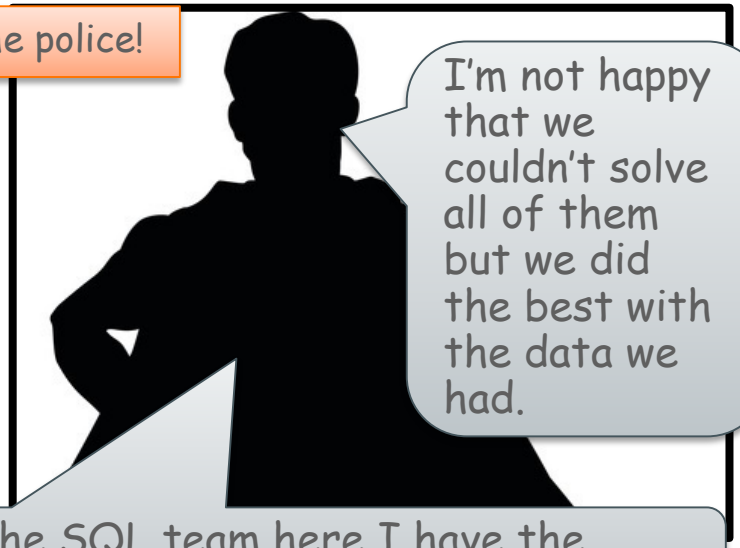
Suspect #3 :
Suspect's details:
Sex: Female
Hair color: Red
Glasses: Yes
Feet size: Small

```
SELECT name
FROM suspects
WHERE sex        = 'Female'
AND   hair_color = 'Red'
AND   glasses    = 'Yes'
AND   scars      = 'No'
AND   feet_size  = 'Small';
```

Number of Results: 2

Suspect details:
Amery Hatfield
Prescott Burch

Now that you have confirmed the results it's time to tell the police!

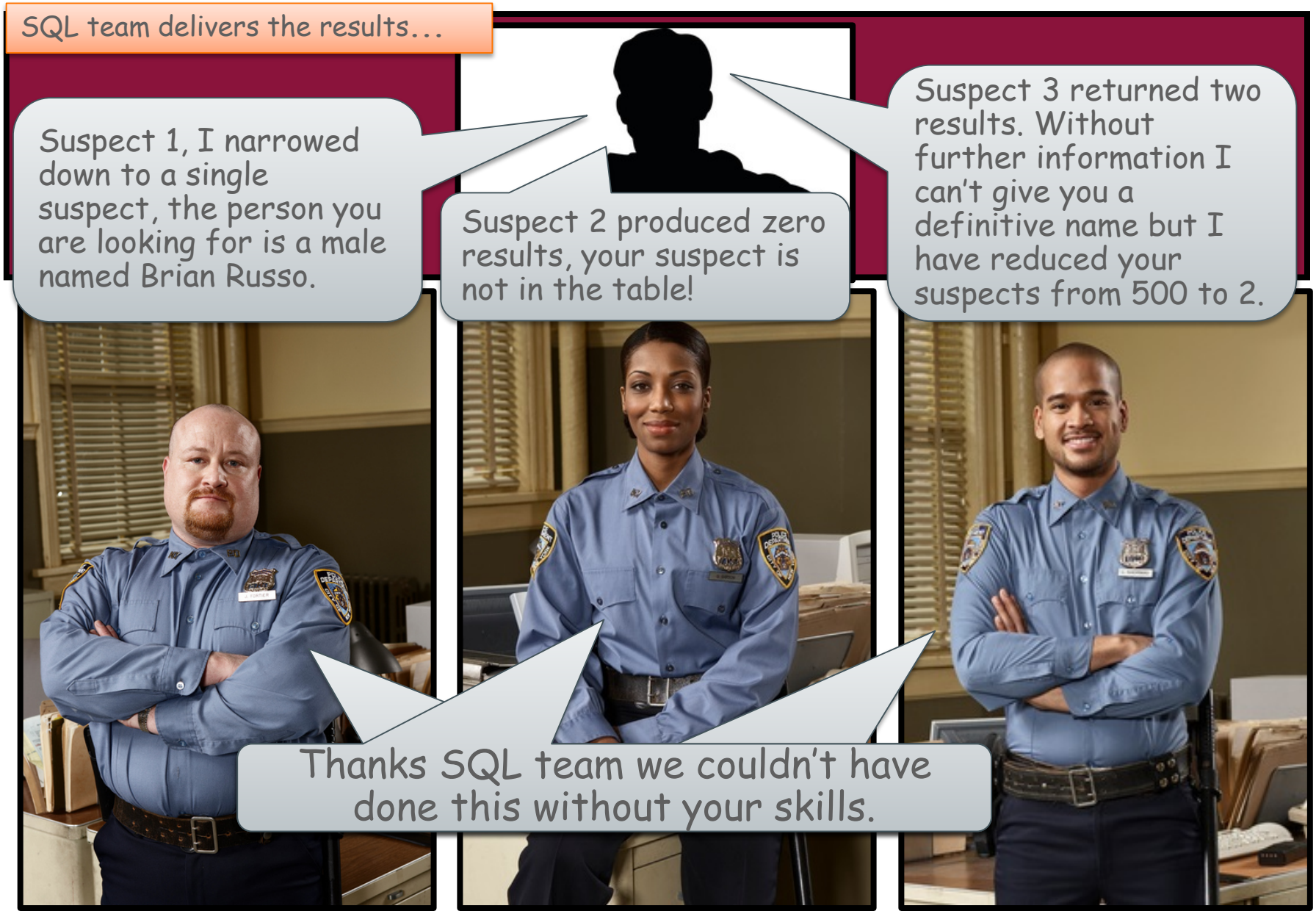


Ring! Ring!

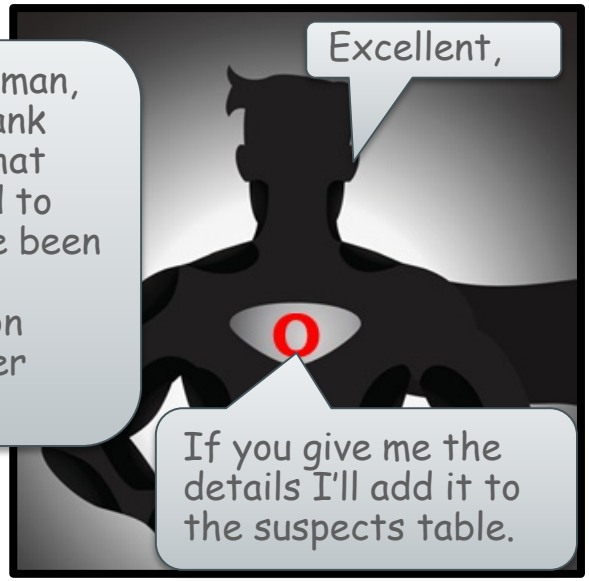
It's the SQL team here I have the results of your open cases.

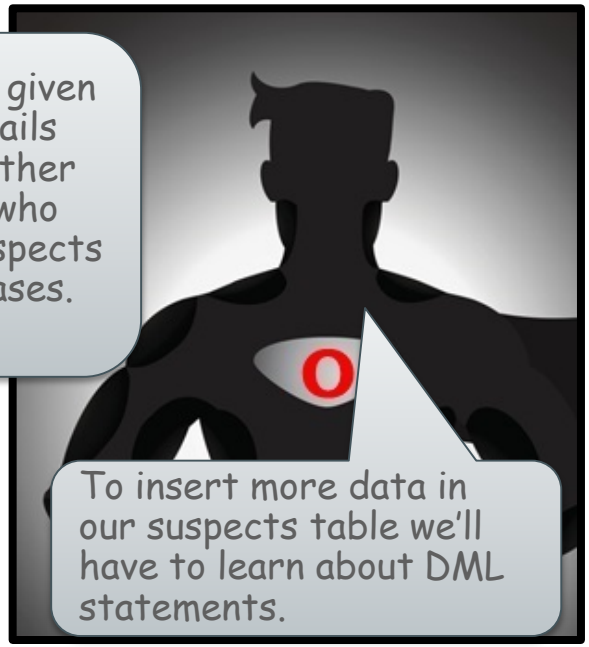
The police officers are anxious to hear the results

SQL team delivers the results...



It's the hotline from police headquarters...





Solve It With SQL

Lesson 2
Introduction to SQL

Part 8 – Data Manipulation Language

Data Manipulation Language

Time for some more theory...

Data manipulation language (DML) is a core part of SQL.

When you want to add, update, or delete data in the database, you execute a DML statement.

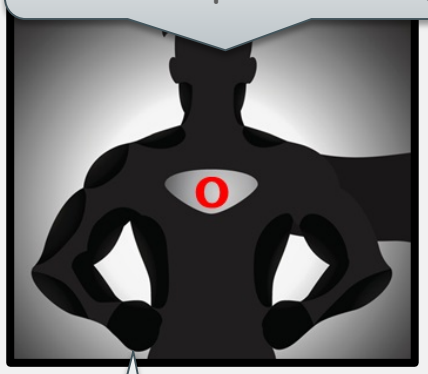
A DML statement is executed when you:

- Add new rows to a table
- Modify existing rows in a table
- Remove existing rows from a table.

In this course we will look at inserting data into a table.

DML: Adding a New Row to a Table

Data manipulation language (DML) is a core part of SQL.



Information in the existing MEMBERS table

MEMBER_ID	FIRST_NAME	LAST_NAME	JOB_ID
103	Alexander	Hunold	IT_PROG
104	Bruce	Ernst	IT_PROG
107	Diana	Lorentz	IT_PROG
124	Kevin	Mourgos	ST_MAN
141	Trenna	Rajs	ST_CLERK

Data to be entered

105	Martin	Keir	AD_VP	90	2200

When we insert a row it is added to the data that is already there.

The order of the data in a table is not important.

MEMBER_ID	FIRST_NAME	LAST_NAME	JOB_ID
103	Alexander	Hunold	IT_PROG
104	Bruce	Ernst	IT_PROG
105	Martin	Keir	AD_VP
107	Diana	Lorentz	IT_PROG
124	Kevin	Mourgos	ST_MAN

Inserted new row in the MEMBERS table.

DML: INSERT Statement Syntax - Implicit

Add a row to a table by using the INSERT statement.

You can insert a row containing values for every column.

```
INSERT INTO members
VALUES (105, 'Martin', 'Keir', 'AD_VP', 90, 2200);
```

This is known as an implicit insert statement.

When you insert a row containing values for each column into a table enclose character and date values within single quotation marks.

Numeric values are not enclosed in single quotes!!

With this insert syntax, only one row can be inserted at a time.

DML: INSERT Statement Syntax - Explicit

You can also use an explicit INSERT statement.

This means that you have to list the columns that you want to include in the INSERT clause.

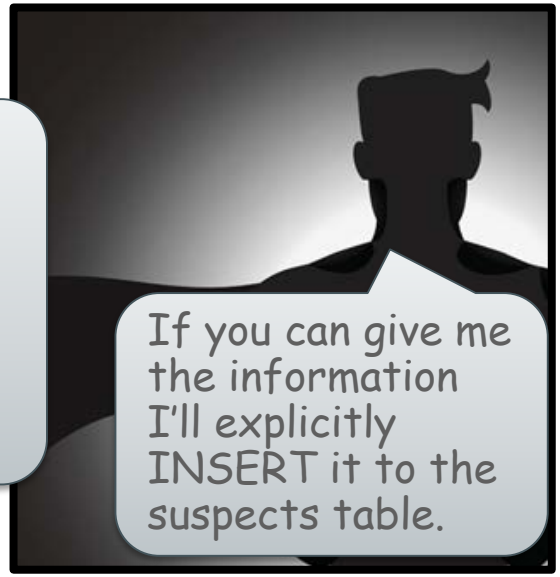
This means that you can INSERT a value for every column.

```
INSERT INTO members (member_id, first_name,
                     last_name, job_id,
                     department_id, salary)
VALUES (106, 'Robyn', 'Victoria', 'ST_CLERK', 50, 1700);
```

You can also only add values for some columns but not others. Beware of columns that need a value (NOT NULL) as not providing one would cause an error.

```
INSERT INTO members (member_id, first_name, last_name)
VALUES (108, 'Rachel', 'Ann');
```

Let's add the information about the criminals to the suspects table.



Criminal #1 :
Name: Josephine Chill
Sex: Female
Age: 27
Height: Tall
Hair color: Red
Eye Color: Green
Facial hair: Yes
Tattoos: Yes
Glasses: No
Scars: Yes
Feet size: Large

Criminal #2 :
Name: Harley Jones
Sex: Male
Age: 42
Height: Medium
Hair color: Brown
Eye Color: Blue
Facial hair: No
Tattoos: No
Glasses: No
Scars: No
Feet size: Medium

Criminal #3 :
Name: Pat Stevenson
Sex: Male
Age: 19
Height: Short
Hair color: Green
Eye Color: Green
Facial hair: Yes
Tattoos: Yes
Glasses: No
Scars: No
Feet size: Large

For information on what to do with this data go to the next slide. . .

Carrying out the INSERT

The SQL team are set to work adding the new information...

Each suspect has to have their own unique suspect_id. I'll have to give them the next number in the sequence from the ones already used in the table.

Before I run the following explicit INSERT statement I should check with SQLman that I have written it correctly!

```
INSERT INTO suspects(suspect_id, name, sex, age, height,
                     hair_color, eye_color, facial_hair,
                     tattoos, glasses, scars, feet_size)
VALUES (501,'Josephine Chill','Female',27,'Tall','Red',
        'Green','Yes','Yes','No','Yes','Large');
```

That's excellent, no errors there!! You can go ahead and run it.

Do the same for the other two sets of data using 502 and 503 as the suspect id's.

Reviewing old cases

We have more data in our table. What can we do with it?

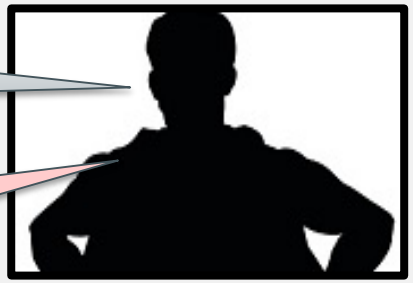
Why did we have to give the suspects a unique suspect id?

It means that each suspect is unique. We can't have two suspects the same because if we try to add two suspect id's that are identical we will get an error!

Now that we have more data in our table we should review some of the cases that didn't return any results earlier.

Suspect 2 returned zero results, we could try that one again.

I'll get the file so we can review the details.



Reviewing old cases cont..

Re-write the query to find suspect 2 or find it through the HISTORY tab in APEX.

The HISTORY tab keeps track of the queries you have previously ran.

We've got a match.

I better contact the police!

Suspect #2 :
Suspect's details:
Sex: Female
Height: Tall
Facial hair: Yes
Glasses: No
Scars: Yes
Tattoos: Yes
Feet size: Large
Age: less than 30

```
SELECT name
FROM suspects
WHERE sex        = 'Female'
AND   height     = 'Tall'
AND   facial_hair = 'Yes'
AND   glasses    = 'No'
AND   scars      = 'Yes'
AND   tattoos    = 'Yes'
AND   feet_size  = 'Large'
AND   age < 30;
```

Number of Results: 1

Suspect details:
Josephine Chill

Let's give the police a nice surprise. . .

Ring! Ring!

Hi SQL team what can I do for you?

It's more what we can do for you.

We re-ran the query for suspect 2 and narrowed it down to a single suspect, the person you are looking for is a female named Josephine Chill.

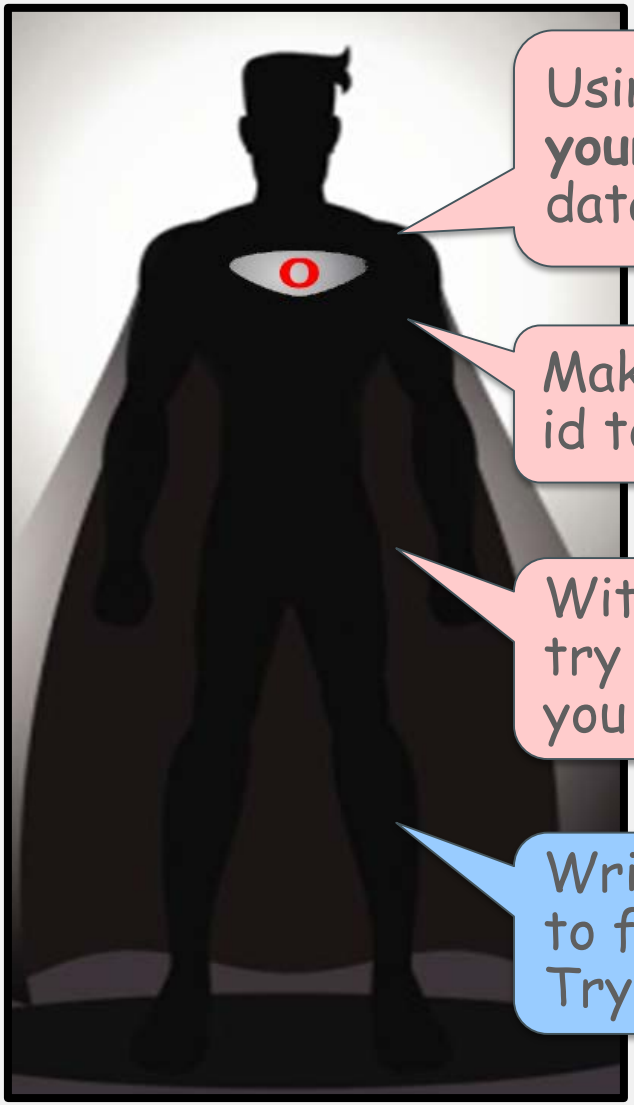
That only leaves Suspect 3 outstanding.

As soon as I get more information I will let you know. We will catch them!

The SQL team have made this city a safer place for it's citizens!!

And so with that the SQL team can relax and be proud of what they have achieved!

Final Challenge! Add yourself to the system



Using an INSERT statement add a record with **your** name and details to the suspects database.

Make sure you assign the next unique suspect id to yourself!

Without using your name as part of the query try to create a SQL statement to return only you as a row.

Write down the SQL statement that you used to find yourself!!
Try to do it with as few conditions as possible!

Summary

In this lesson, you should have learned how to:

- Describe what SQL is.
- Create SQL Statements
- Identify Conditions
- Use Conditions with Logical Operators
- Understand the Rules of Precedence
- Use Comparison Operators
- Understand Data Manipulation Language
  - Use Insert statements.


ORACLE
ACADEMY