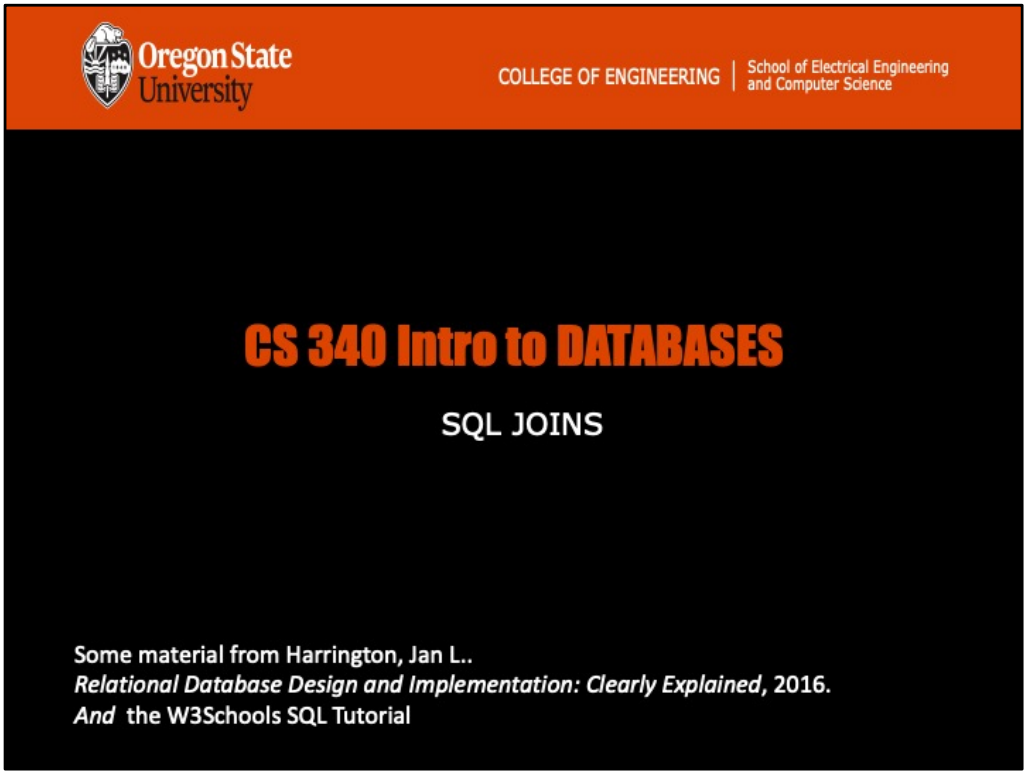
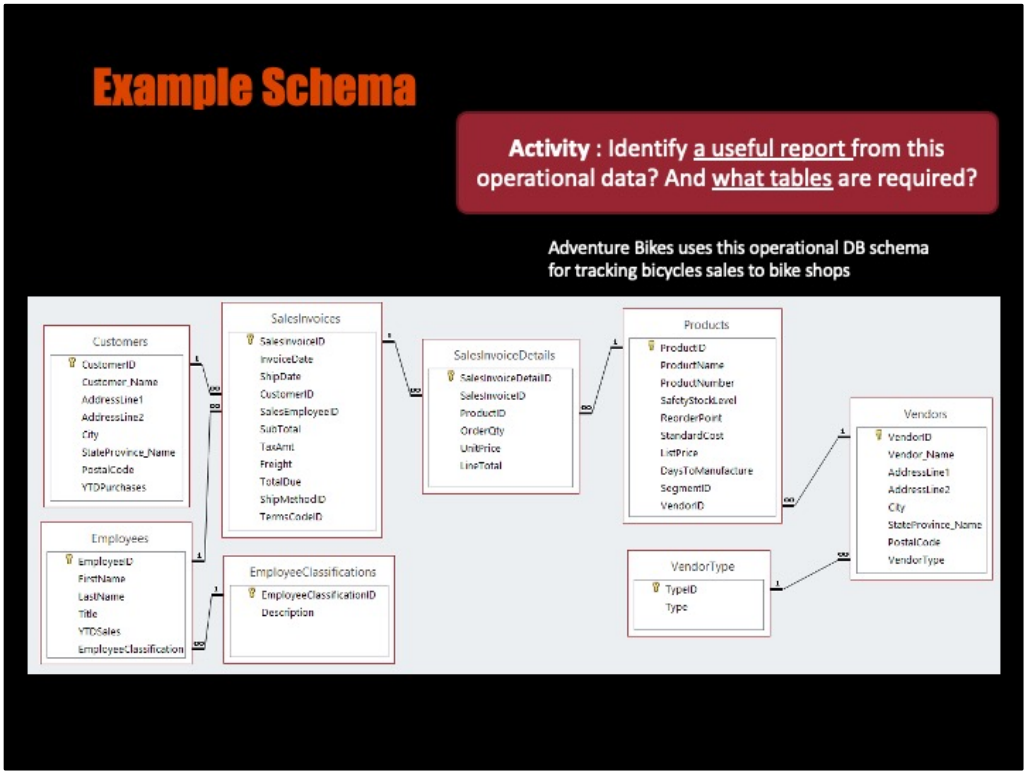


Some of this material comes from the course textbook chapter 17 & 19, which you can read for free from the OSU Library.

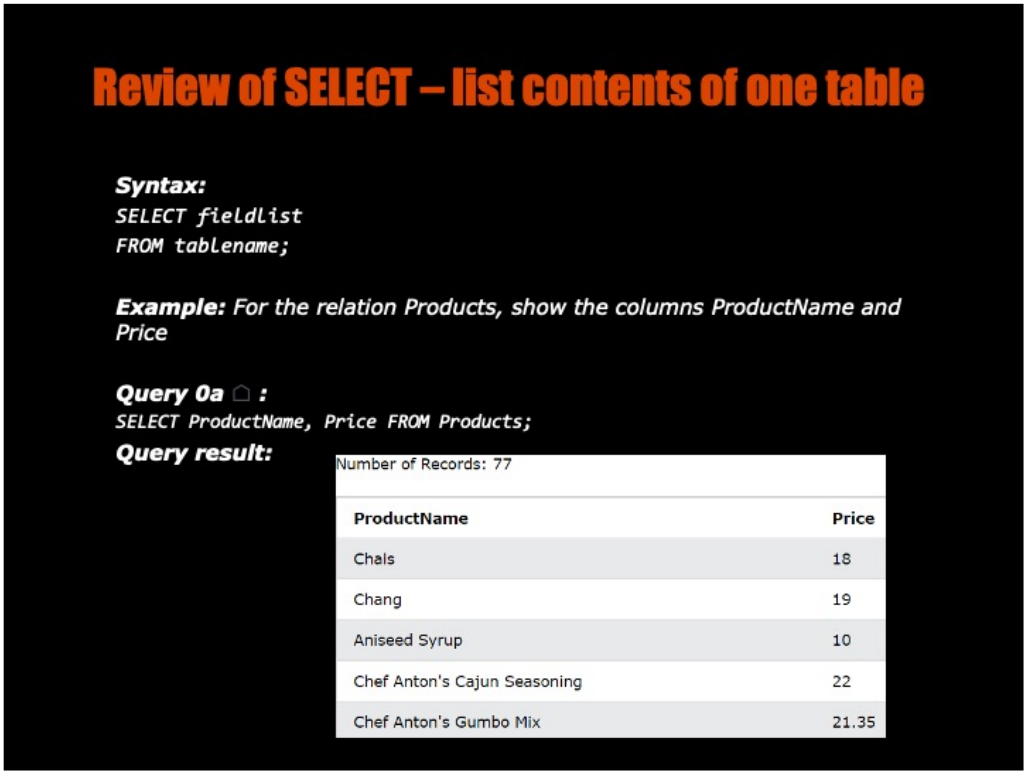
Harrington, Jan L.. *Relational Database Design and Implementation : Clearly Explained*, Elsevier Science & Technology, 2016. *ProQuest Ebook Central*, http://ebookcentral.proquest.com/lib/osu/detail.action?docID=4509772

As well as the W3Schools SQL Tutorial at https://www.w3schools.com/sql/default.asp –you are encouraged to follow along with the activities available at this link.



- This is an example operational database schema used by Adventure Bikes who wants to record sales of bicycles.
- In addition to recording sales, Adventure Bikes may benefit from generating various useful reports from this operational data
    - For example **an invoice** –the items a customer bought and how much the entire order cost, could be produced with **all four tables** in this schema
- **Activity**:
    - Identify <u>one other useful reports</u>, and <u>the tables</u> contain the data necessary for that report.
    - If your stuck, think about what kinds of things Adventure Bikes may want to know about their customers or products, as a suggestion.

Tip: Knowing what reports your customers will want can help you better design the database to support their needs.



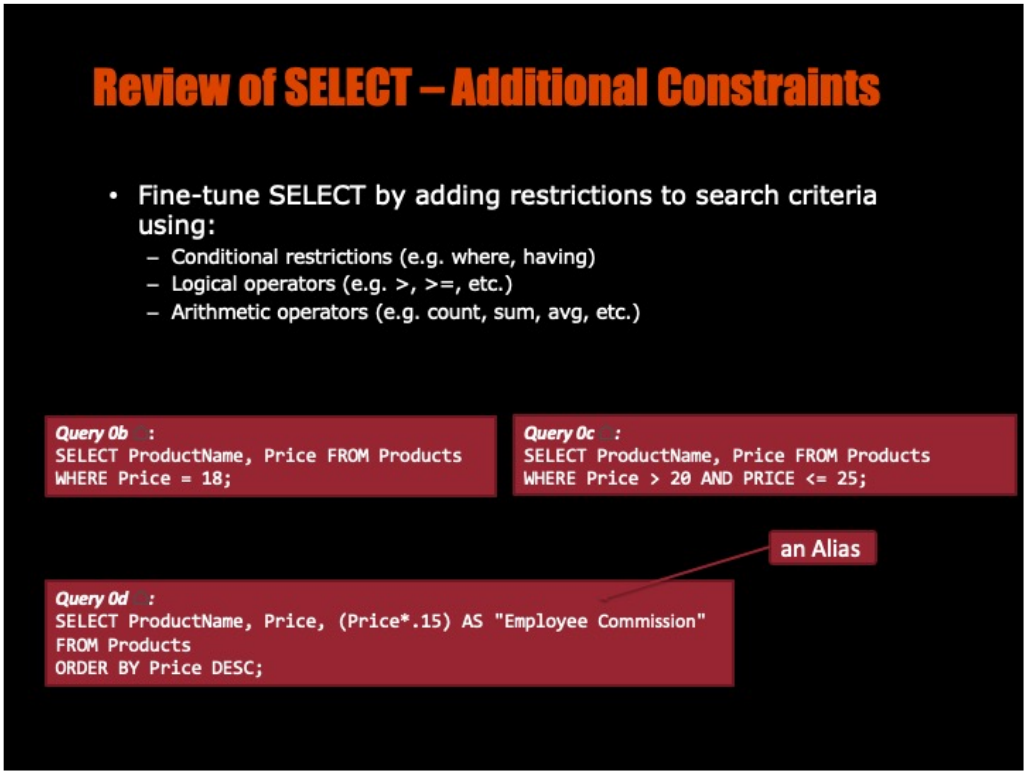
Review of SELECT

Recall that the SELECT statement is used to select data from a database. The data returned is stored in a result table, called the result-set.

You can refresh yourself with the SELECT at the following https://www.w3schools.com/SQl/trysql.asp?filename=trysql_select_all

And run the following command

```
SELECT ProductName, Price FROM Products;
```



The WHERE clause is used to filter records. It is used to extract only those records that fulfill a specified condition. It can be used with operators such as =, >, <> and more.

Practice at the following
https://www.w3schools.com/SQl/trysql.asp?filename=trysql_op_equal_to

SELECT ProductName, Price FROM Products
WHERE Price = 18;

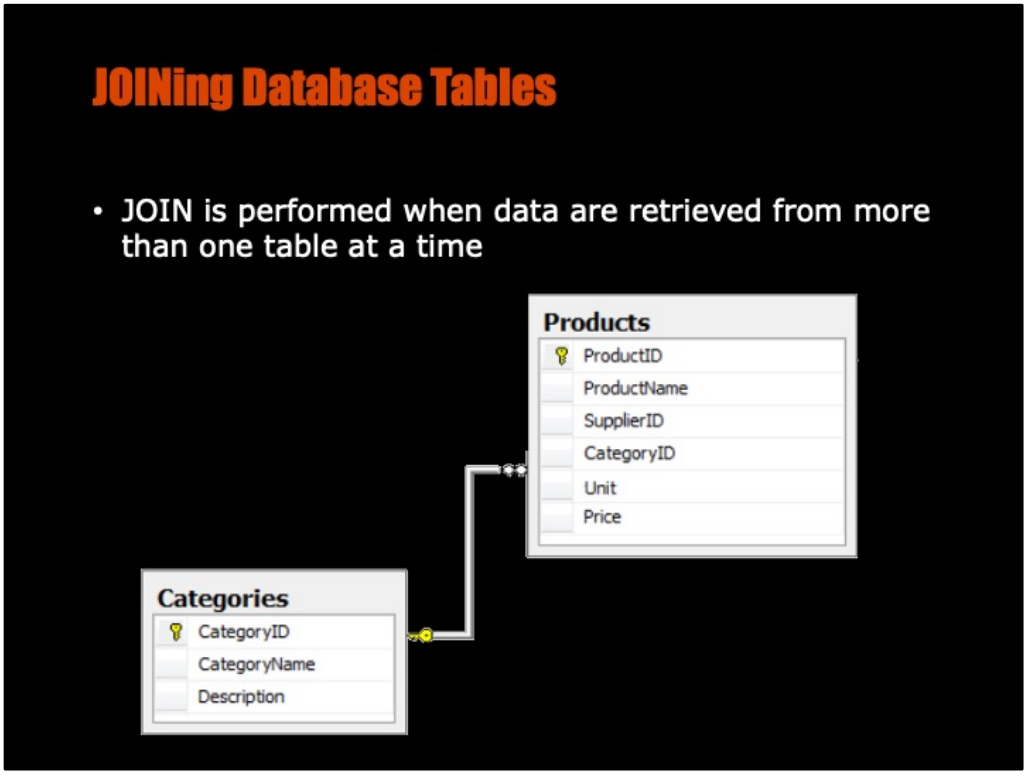
SQL aliases are used to give a table, or a column in a table, a temporary name. Aliases are often used to make column names more readable. An alias only exists for the duration of that query. An alias is created with the AS keyword.
https://www.w3schools.com/SQl/sql_alias.asp

SELECT ProductName, Price, (Price*.15) AS "Employee Commission"
FROM Products
ORDER BY Price DESC;

The GROUP BY statement groups rows that have the same values into summary rows. It is often used with aggregate functions like COUNT(), MAX(), MIN(), SUM(), and AVG() to group the result-set by one or more columns. The HAVING clause is used to filter, because the WHERE keyword cannot be used with aggregate functions. Again, take a

minute and try these yourself:

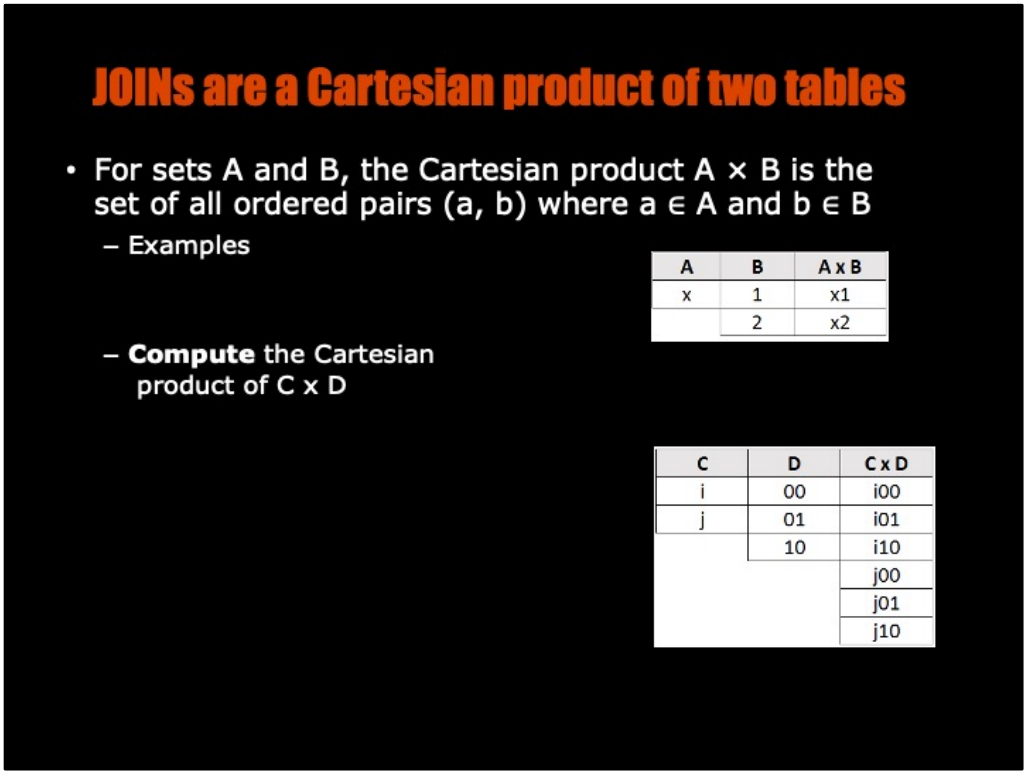
https://www.w3schools.com/SQl/sql_groupby.asp
https://www.w3schools.com/SQl/sql_having.asp



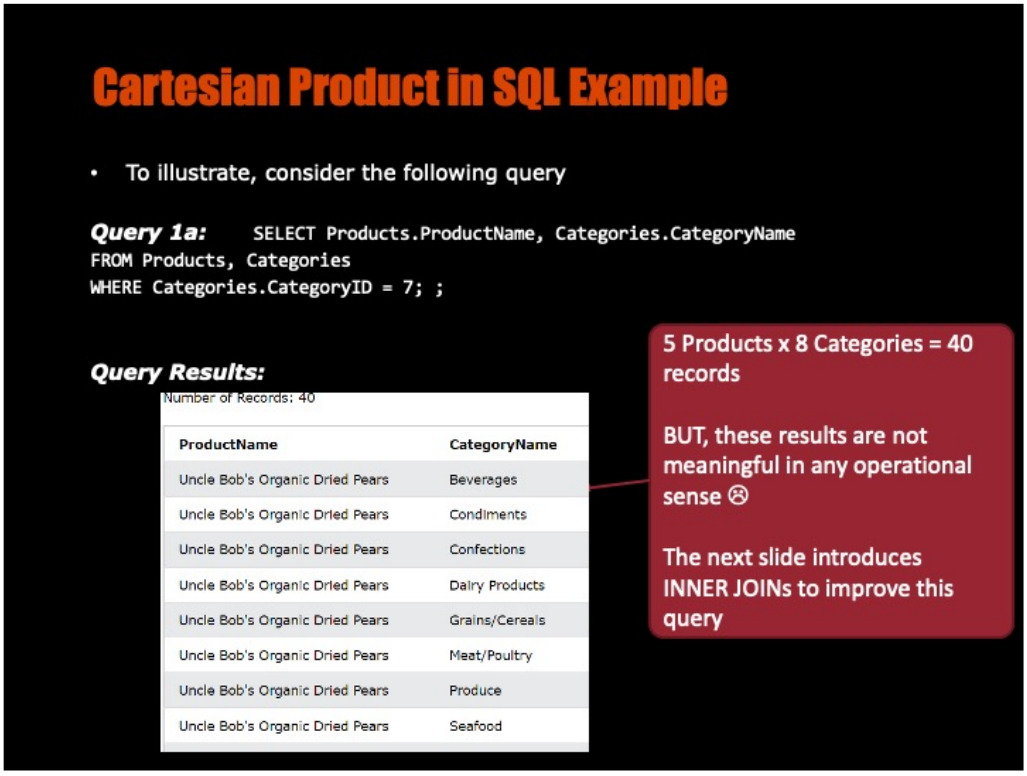
Recall that a JOIN clause is used to combine rows from two or more tables, based on a related column between them.

In the above example, if we wanted to view each Product and its associated Category we must specify these tables are being joined where Products.CategoryID = Categories.CategoryID

The Table.Column notation is used to resolve ambiguity when columns have identical names.



In essence, JOINs are a Cartesian product of two tables.



Activity: you can see what the Cartesian product looks like yourself by going to the following

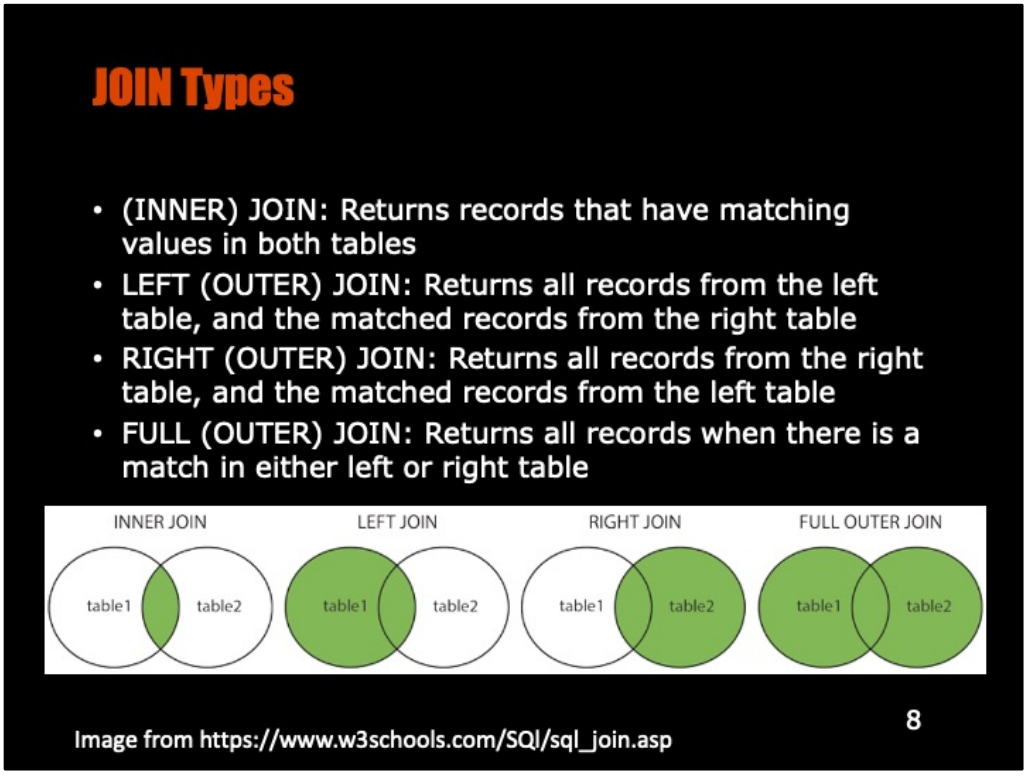
https://www.w3schools.com/SQl/sql_join.asp and clicking on the Try It Yourself button (or go directly to the Try-SQL editor at https://www.w3schools.com/SQl/trysql.asp?filename=trysql_select_join)

Modify the query in this online SQL editor to be as follows (notice that there is NO JOIN):

```
SELECT Products.ProductName, Categories.CategoryName
FROM Products, Categories
WHERE Products.CategoryID = 7;
```

The Cartesian product is 5 Products x 8 Categories = 40 records.

BUT, these results are not meaningful in any operational sense ☹

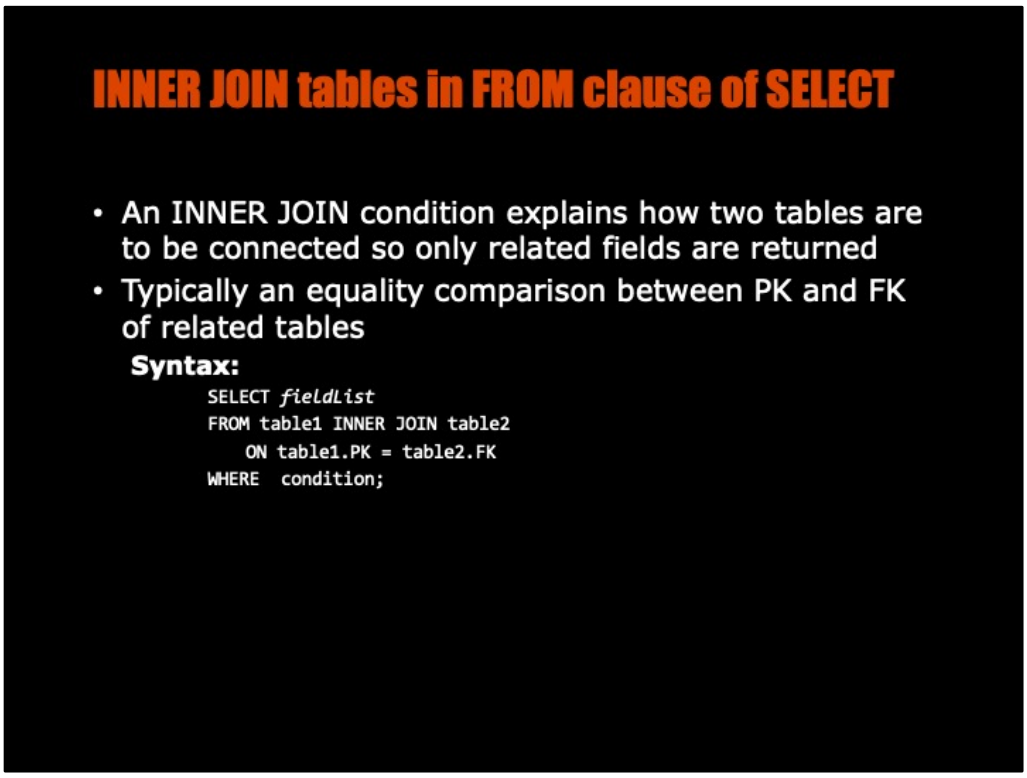


JOIN Types

(INNER) JOIN: Returns records that have matching values in both tables
LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

In practice, the INNER JOIN is the most common type.
There are times when the OUTER JOINS are useful, but it is rare.

Activity: try these JOINs out yourself at
https://www.w3schools.com/SQl/trysql.asp?filename=trysql_select_join



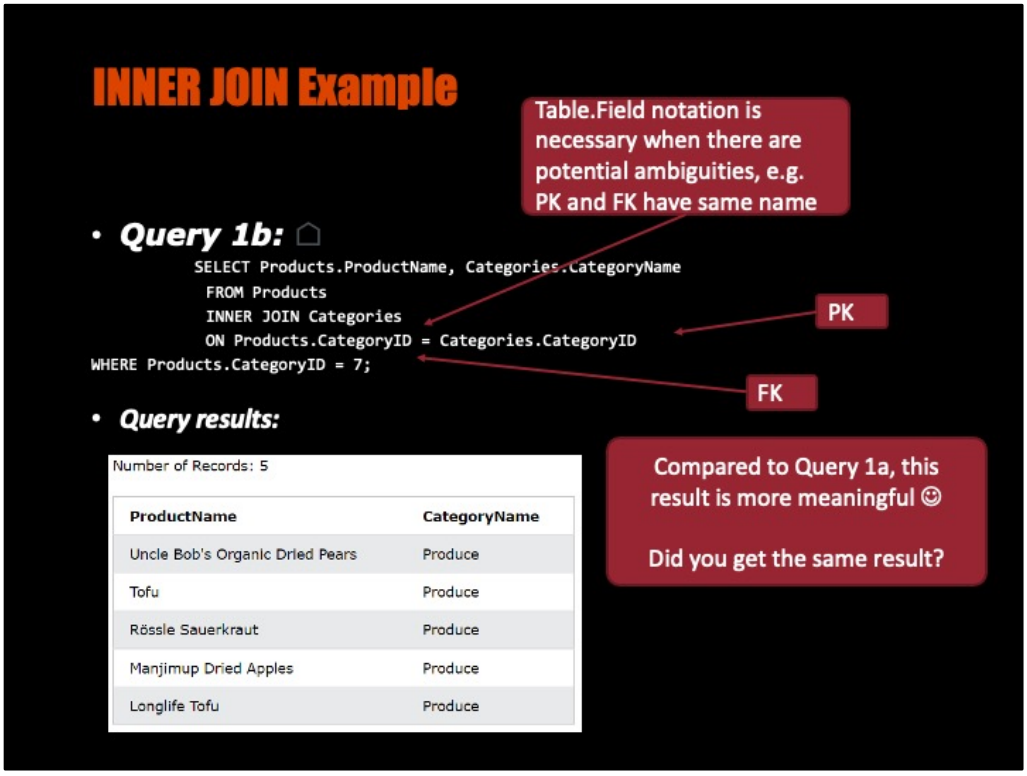
An INNER JOIN condition explains how two tables are to be connected so only related fields are returned

Typically an equality comparison between PK and FK of related tables

Syntax:

```
SELECT fieldList
FROM table1 INNER JOIN
table2
ON table1.PK = table2.FK
WHERE  condition;
```

Activity 6 will give you the opportunity to formulate SQL join queries and many of the concepts we have explored in this lecture.

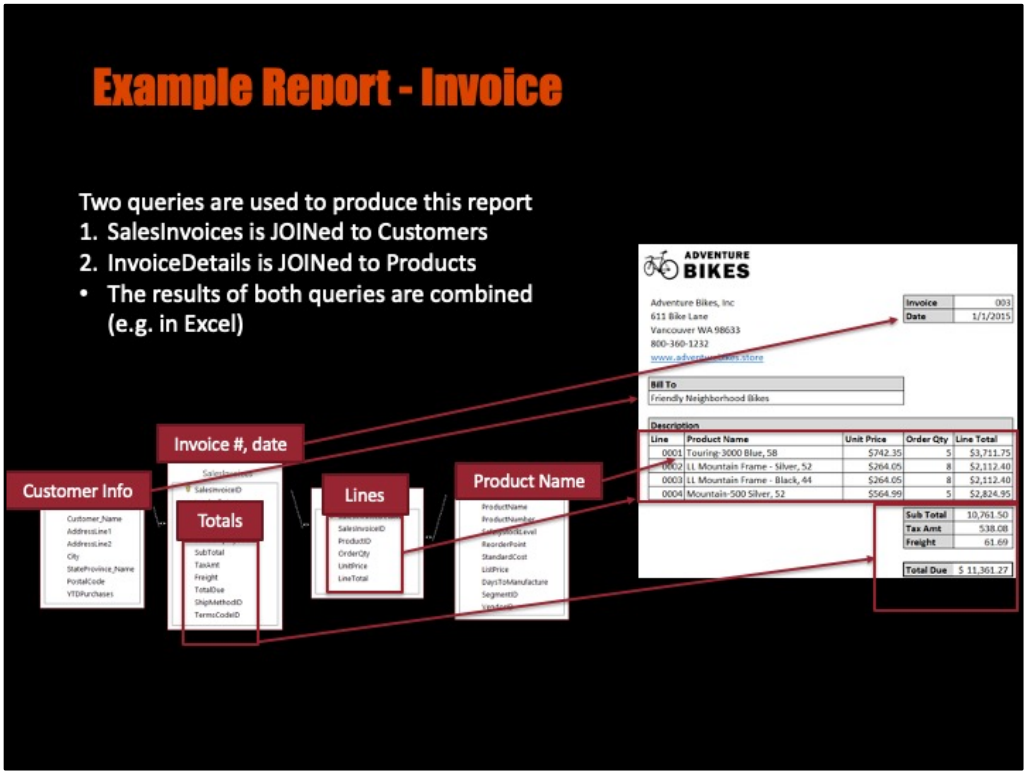


Go back to https://www.w3schools.com/SQl/trysql.asp?filename=trysql_select_join

And try out the following query

```
SELECT Products.ProductName, Categories.CategoryName
FROM Products
        INNER JOIN Categories
        ON Products.CategoryID = Categories.CategoryID
WHERE Products.CategoryID = 7;
```

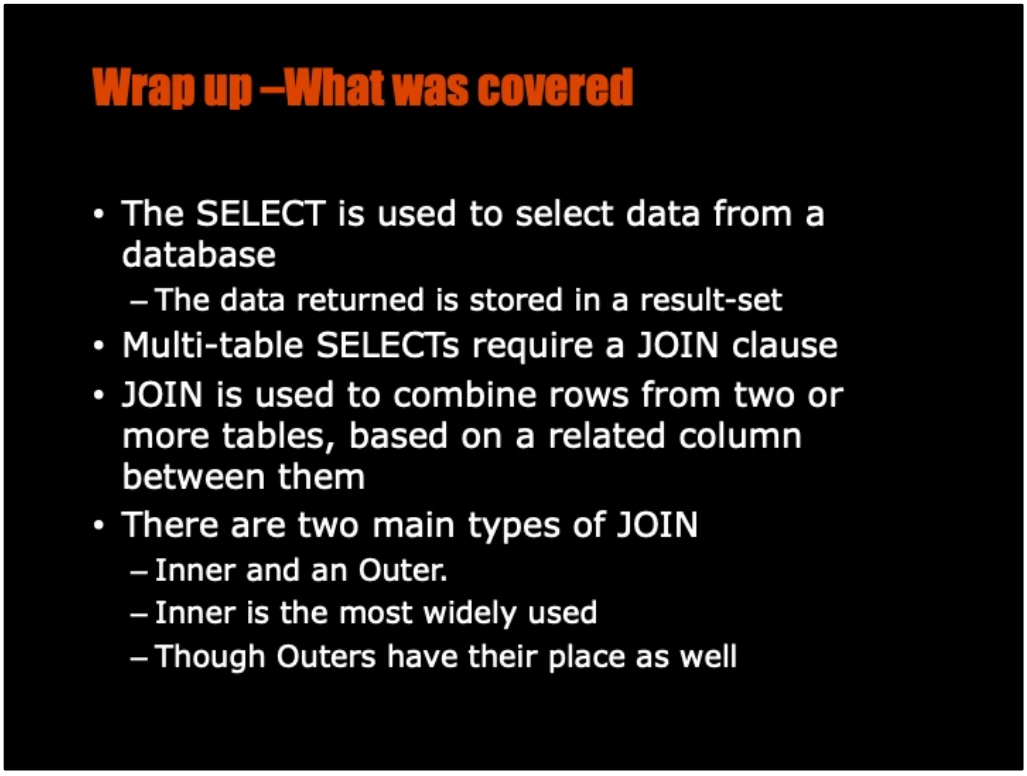
Compared to the previous Query, this result is more meaningful ☺
Did you get the same result?



This invoice is actually a report that was made from two different queries. The results of the query were put into a reporting tool (e.g. Excel) to make it pretty. We are only focusing on how to write the queries, not how to format them into a report.

1. To produce the report header with customer info and invoice totals, the SalesInvoices table is JOINed to Customers in the first query
2. To produce the report body with line details and product names, InvoiceDetails is JOINed to Products in a second query

Activity 6 will give you the change to formulate a set of queries that might be used to make this report.



Summary

The SELECT is used to select data from a database

The data returned is stored in a result-set

Multi-table SELECTs require a JOIN clause

JOIN is used to combine rows from two or more tables, based on a related column between them

There are two main types of JOIN

Inner and an Outer.

Inner is the most widely used

Though Outers have their place as well