

SQL and Database Management

BEGINNING AND INTERMEDIATE



What to expect

- Basic terminology
- Relational database design
- Database creation
- Table edits
- Simple queries

Quick poll of experience

Kate Amorella Proff

- 6 years in Institutional Research
 - 4 with Texas Woman's University
 - 2 with Texas State University
- BS in Computer Science
 - ...yet mainly self-taught
- Interests
 - Technology
 - “Lifhacking”
 - Softball
 - Pretending to read books for book club

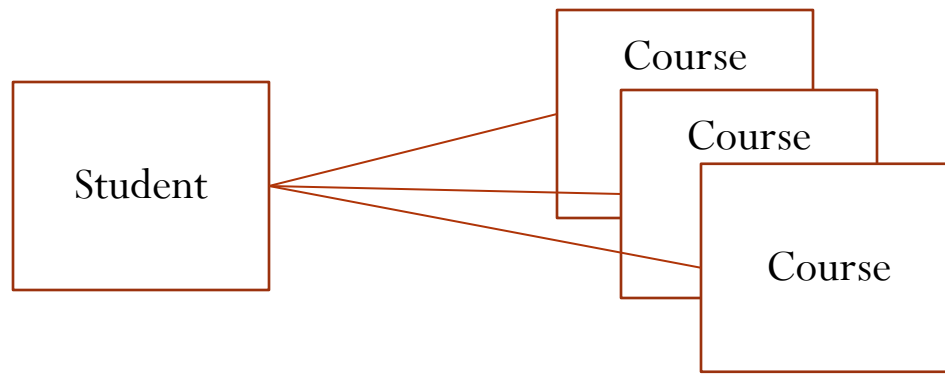
Tracy Stegmair

- X years in Institutional Research
 - X with Texas Woman's University
 - X with Richland College
 - Now back with TWU
- BS in History
- Interests

Terminology

What is a relational database?

- A collection of data items
- Stored in tables
 - Organized by columns and rows
- Tables can be joined by relations
- Basically, a relational database is capable of representing relationships between different sets of data



What is a DBMS?

- Database Management System
- Program used for storing, modifying, and extracting data from a relational database
- Popular DBMS's include:
 - Oracle
 - IBM
 - Microsoft
 - MySQL

What is SQL?

- Structured Query Language
- Language for “talking” to a DBMS
- Used in basic operation
 - Create, insert, update, delete
- Used for describing the data
 - select, from, where

Data Types

- Description of field contents
- Will vary based on the DBMS
- String
 - Text, VARCHAR, CHAR
- Number
 - Integer, Double, Float, Decimal
- Date/Time
 - Date, Datetime, Time, Year

http://www.w3schools.com/sql/sql_datatypes.asp

Table Relationships

- One-to-One
 - One student has one birthdate
- One-to-Many
 - One student has many addresses
- Many-to-Many
 - Many instructors teach many courses

Join Types

- Join (Inner Join)
 - Return rows where there is at least one match in both tables
- Left Join
 - Return all rows from the left table, regardless of a match on the right table
- Right Join
 - Return all rows from the right table, regardless of a match on the left table
- Full Join
 - Return all rows where there is a match in either table

<http://net.tutsplus.com/tutorials/databases/sql-for-beginners-part-3-database-relationships/>

Resources

- <http://www.1keydata.com/sql/sql.html>
- <http://www.w3schools.com/sql/default.asp>
- <http://plsql-tutorial.com/index.htm>
- <http://www.techonthenet.com/sql/index.php>

Table of Contents

Access Basics	1
Access Query Tools	2
The Development Sandbox.....	4
Accessing your MySQL Database	7
SQL Syntax.....	13
SQL Developer Tips & Tricks	19

Access Databases

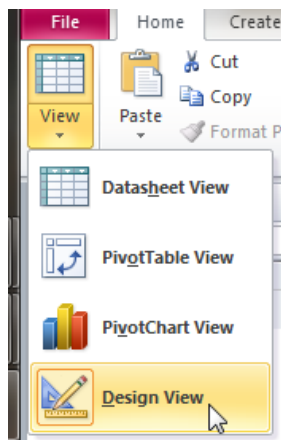
Access Basics

Open a table

- Double-click on a table in the Tables pane

View Data Types

- In the ribbon, click the arrow under View,
- Select Design View



Create a Table

- In the ribbon, select Create > Table
- Go to the Design View to add fields
- Enter data in the Datasheet View

Import Data

- In the ribbon, select External Data
- Choose the data type you would like to import
- Follow the wizard to import your data into a new table, append data, or link to data

Access Query Tools

Query Wizard

- In the ribbon, select Create > Query Wizard
- Follow the wizard to create a query
 - Note: To join tables with the Query Wizard, you must set up database Relationships first

Query Design

- In the ribbon, select Create > Query Design
- Choose the tables you would like to query
- Join your tables
 - Select your join type by double-clicking on the join



- Select the fields you want to view

- Either double-click the values in the table list, or
- Choose the field and table from the drop-down in the criteria section
- Define your selection criteria in the Criteria row

Field:	FIRST_NAME	LAST_NAME	ACADEMIC_PERIOD	LATEST_DECISION
Table:	PERSON	PERSON	APPLICATIONS	APPLICATIONS
Sort:				
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			201210	"AC"
or:				

- Click Run, in the ribbon, to view your output

SQL View

- In the ribbon, select Create > Query Design
- Choose your tables or click cancel
- In the ribbon, make sure you are on the Design tab
- Click the arrow under View
- Select SQL View
- Enter your query

Query1
<pre>SELECT PERSON.FIRST_NAME, PERSON.LAST_NAME, APPLICATIONS.ACADEMIC_PERIOD, APPLICATIONS.LATEST_DECISION FROM APPLICATIONS INNER JOIN PERSON ON APPLICATIONS.ID = PERSON.ID WHERE (((APPLICATIONS.ACADEMIC_PERIOD)=201210) AND ((APPLICATIONS.LATEST_DECISION)="AC"));</pre>

- Click Run, in the ribbon, to view your output

Export Data

- In the ribbon, select External Data
- Choose the data type you would like to export to
- Follow the wizard to import your data into a new table, append data, or link to data

MySQL Databases

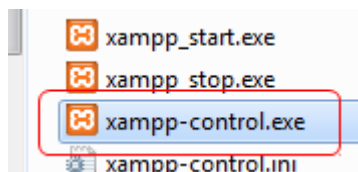
The Development Sandbox

XAMPP

XAMPP is a cross-platform open-source web server which contains the Apache HTTP Server, a MySQL database, and interpreters for PHP and PERL. We will use this to allow use to test our work on our local machines. The version we are using is the portable version. If you would like more information on XAMPP and other development server tools, feel free to contact Kate after this workshop.

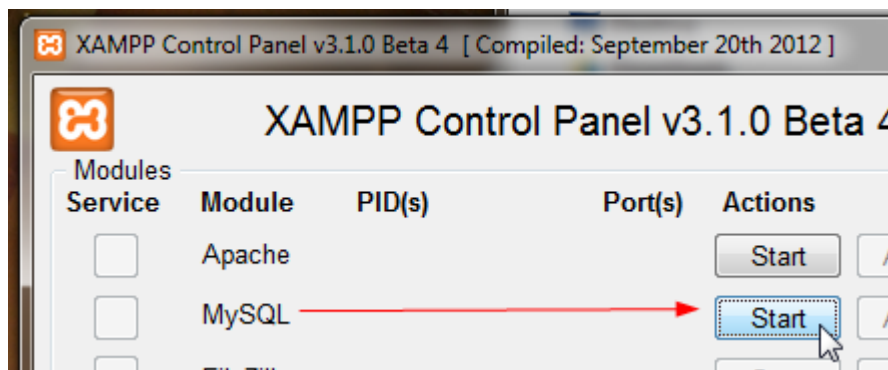
Starting XAMPP

- Open the **Workshop** folder on your flash drive
- Open the **xampp-portable** folder
- Find **xampp-control.exe**
- Double-click **xampp-control.exe** to run

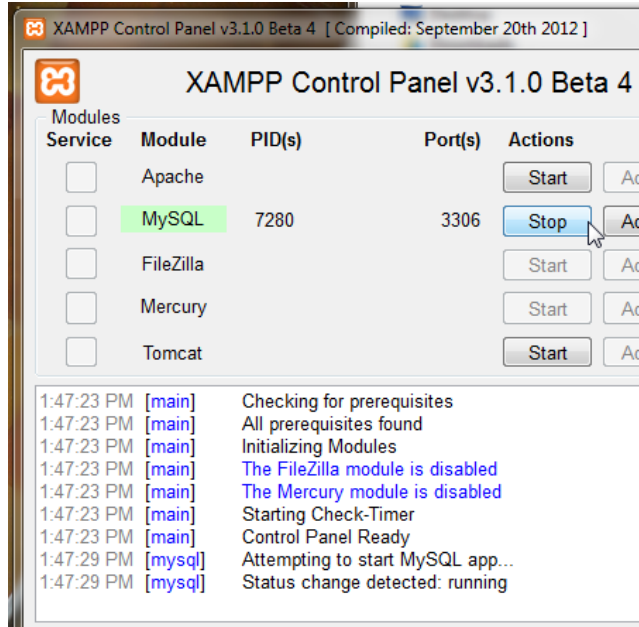


Starting the Services

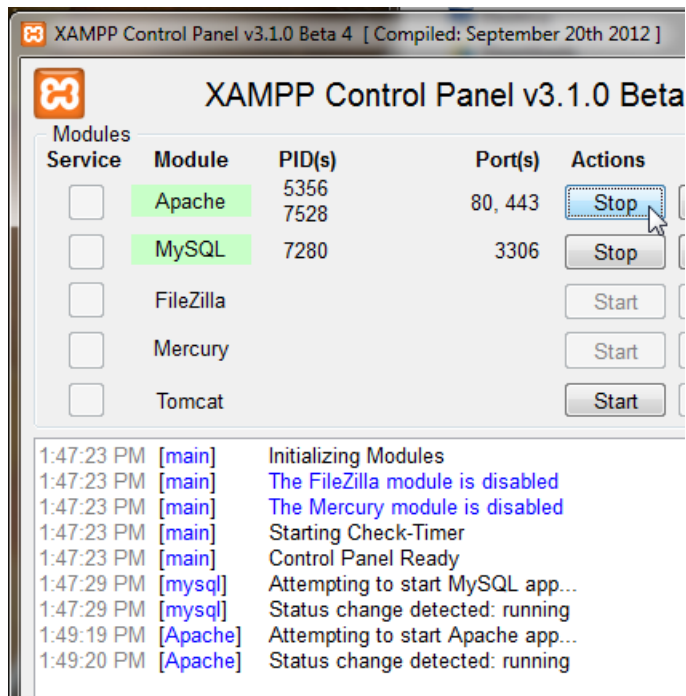
- In the XAMPP Control Panel, click Start next to the MySQL service



- When the service is running, you will see a PID and Port 3306 displayed



- Next, start the Apache web server by clicking the Start button next to Apache
- When the service is running, you will see PIDs and Ports populated



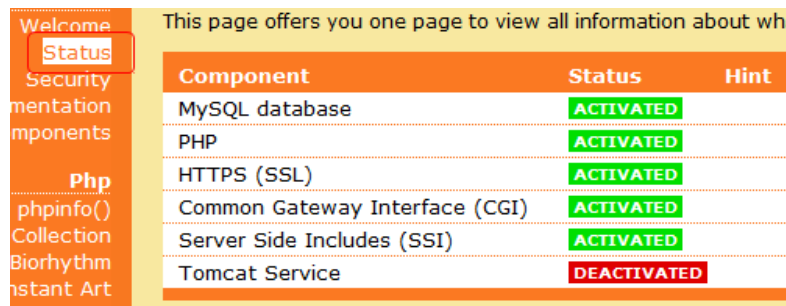
- You can now minimize or “X” the XAMPP window

Test the Services

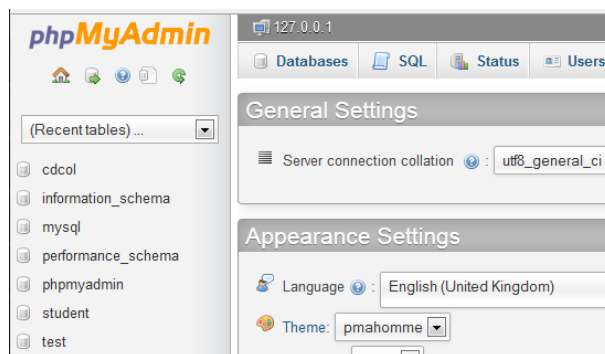
- In your web browser, type localhost into the URL field (address bar)
- You should be greeted with this screen:



- Click the Status link to see that all the required services are running



- Click phpMyAdmin to see that MySQL is running



- Close your web browser

If you would like more information on phpMyAdmin, please contact Kate after the workshop.

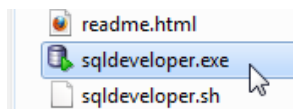
Accessing your MySQL Database

SQL Developer

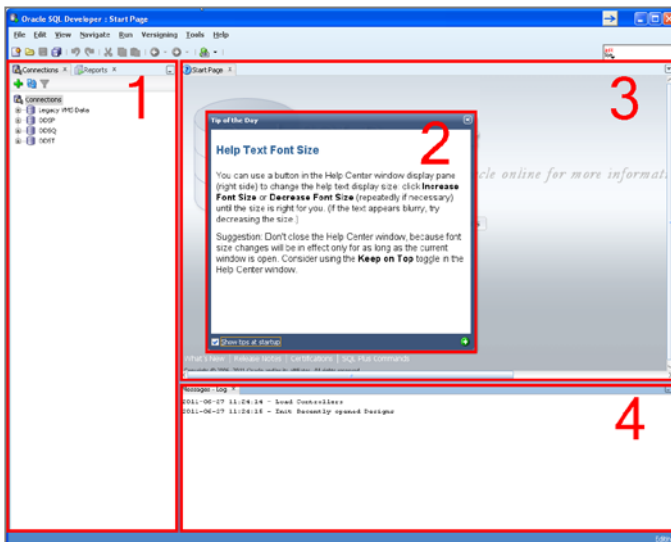
SQL Developer is a free integrated development environment (IDE) for working with SQL in databases. It is a product of the Oracle Corporation. For information on different IDEs, contact Kate.

Starting SQL Developer

- Open the **Workshop** folder on your flash drive
- Open the **sqldeveloper** folder
- Find **sqldeveloper.exe**
- Double-click **sqldeveloper.exe** to run





- SQL Developer will open and look similar to:

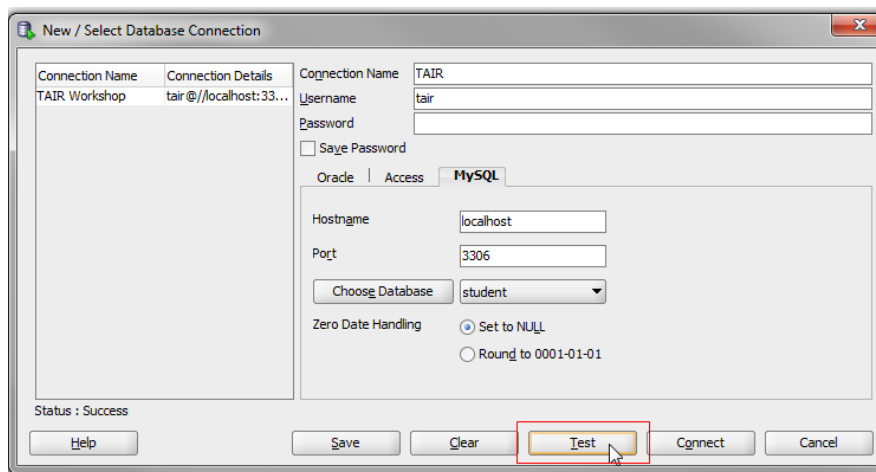


1. Left-panel – contains connections, reports, and the file navigator. **2. Tips box** – can be disabled by unchecking *Show tips at startup*. **3. Right-panel** – contains welcome screen, editor, and data view. **4. Messages-Log** – information on the processes run by SQL Developer. Best to close or minimize while editing SQL statements.

- **Close** the Tips box for now

Setting Up Database Connections

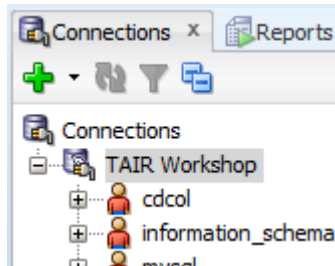
- Create a new database connection by doing one of the following:
 - Click the **New/Select Database Connection** icon 
 - Selecting **File > New > Database Connection**
 - Click the **Create a New** icon , and select **Database Connection**
- In the New/Select Database Connections window, select the MySQL tab



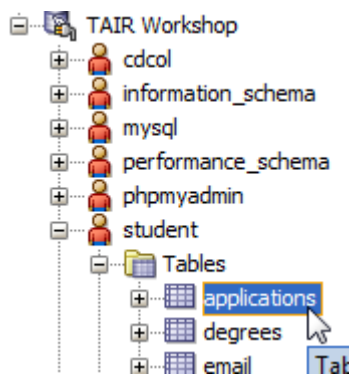
- In the **Connection Name** textbox, enter the name you'd like to associate with your connection database. This can be changed later.
- Enter **tair** for the **Username**
- For the **Hostname**, type **localhost**
- For **Port**, type **3306**
- Click the **Choose Database** button
- Select **student** from the dropdown
- Click **Test**, to test the connection
 - The status of Success should show in the lower right-hand corner
 - If you receive an error, please let Kate or Tracy know
 - For the purposes of this workshop, select Cancel

Querying the Database

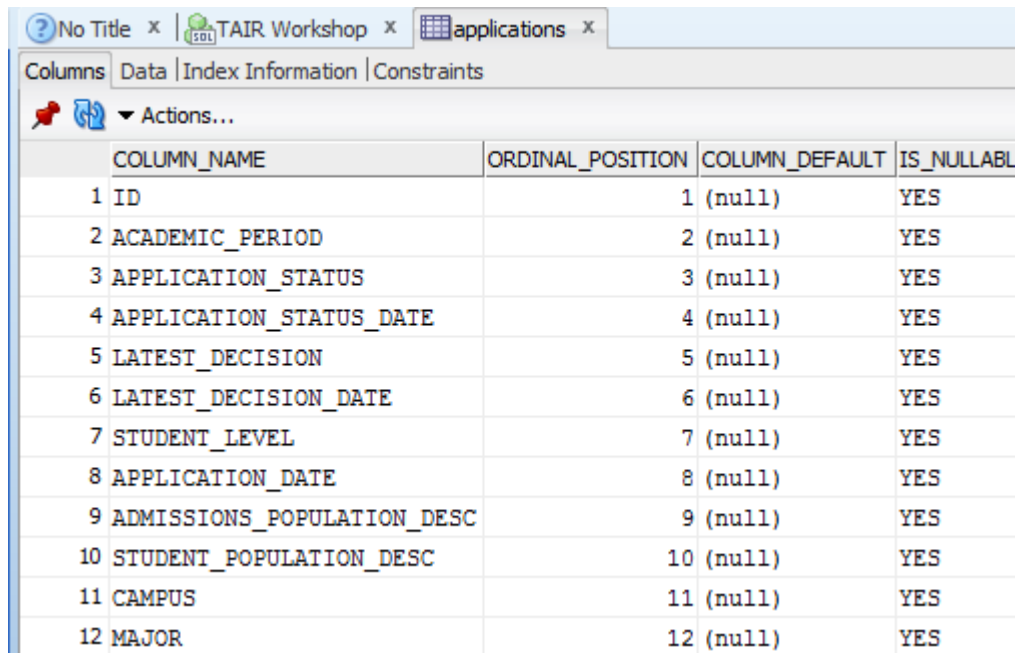
- Expand **TAIR Workshop** by clicking the plus sign next to **TAIR Workshop** or double-clicking directly on **TAIR Workshop** in the Connections tab in the left-panel



- Under **TAIR Workshop**, expand **student**
- Within **student**, you will see many options. We will be using **Tables**.
- Click on **applications**



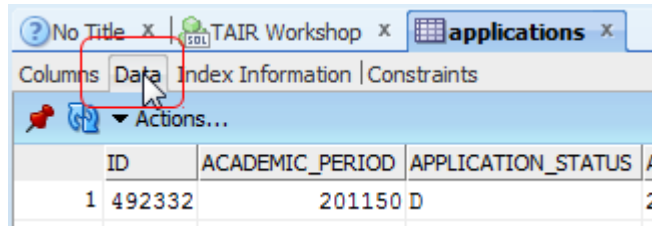
- A **Columns** view of the selected table will appear in the right-panel



	COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE
1	ID	1	(null)	YES
2	ACADEMIC_PERIOD	2	(null)	YES
3	APPLICATION_STATUS	3	(null)	YES
4	APPLICATION_STATUS_DATE	4	(null)	YES
5	LATEST_DECISION	5	(null)	YES
6	LATEST_DECISION_DATE	6	(null)	YES
7	STUDENT_LEVEL	7	(null)	YES
8	APPLICATION_DATE	8	(null)	YES
9	ADMISSIONS_POPULATION_DESC	9	(null)	YES
10	STUDENT_POPULATION_DESC	10	(null)	YES
11	CAMPUS	11	(null)	YES
12	MAJOR	12	(null)	YES

The column view lists each of the variables in the table along with the variable properties

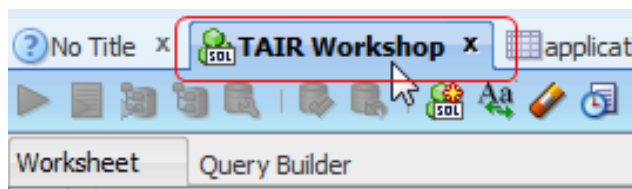
- Click on **Data**, within the selected table tab in the right-panel



	ID	ACADEMIC_PERIOD	APPLICATION_STATUS	A
1	492332	201150	D	2

The data contained with the table will appear in the panel

- Open a blank **SQL Worksheet** by clicking on the TAIR Workshop tab in the right-panel




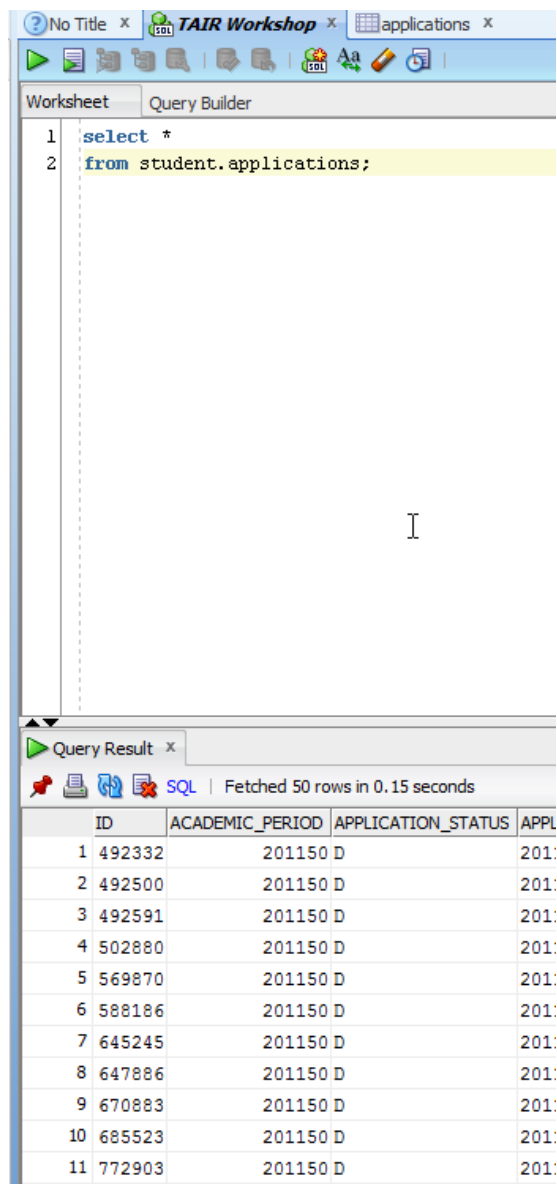
- If there is no open tab, click the **SQL Worksheet** icon to create a new one
- From within an **SQL Worksheet**, you can use SQL to **query** the database



- Type the following in the SQL Worksheet:

```
SELECT *  
FROM student.applications;
```

- To run the query, click the **Run Statement** icon  or press **F9**
- The **Select Connection** window may appear. If so, select which database you would like to query and click OK
- The results of your query will appear in the **Query Result** window below the SQL Worksheet



The screenshot shows a software interface with a SQL Worksheet and a Query Result window. The SQL Worksheet contains the following query:

```
1 select *  
2 from student.applications;
```

The Query Result window displays the following table:

	ID	ACADEMIC_PERIOD	APPLICATION_STATUS	APPL
1	492332	201150	D	201:
2	492500	201150	D	201:
3	492591	201150	D	201:
4	502880	201150	D	201:
5	569870	201150	D	201:
6	588186	201150	D	201:
7	645245	201150	D	201:
8	647886	201150	D	201:
9	670883	201150	D	201:
10	685523	201150	D	201:
11	772903	201150	D	201:

- Find and click on the **applications** table in the Connections tab in the left-panel
- The **applications** table will appear in the right-panel
- Click **Data** within the **applications** tab


The screenshot shows a window titled 'No Title' with tabs for 'TAIR Workshop' and 'applications'. The 'Data' tab is selected, displaying a table with the following data:

	ID	ACADEMIC_PERIOD	APPLICATION_STATUS	APPLICATIO
1	492332	201150	D	2011-04-
2	492500	201150	D	2011-04-
3	492591	201150	D	2011-04-
4	502880	201150	D	2011-04-

- Now click back to the **TAIR Workshop** tab

The screenshot shows a 'Query Result' window with the following data:

	ID	ACADEMIC_PERIOD	APPLICATION_STATUS	APPLICATIO
1	492332	201150	D	2011-04-
2	492500	201150	D	2011-04-
3	492591	201150	D	2011-04-
4	502880	201150	D	2011-04-
5	569870	201150	D	2011-04-
6	588100	201150	D	2011-04-

- You will notice that the data are the same, because our query searched for all records within the applications table
- To save the **query**, do one of the following:
 - Go to **File > Save**
 - Click the **Save** icon 
 - Press **Ctrl+S**

Note: Saving the query saves the SQL statements, not the query results. To save the query results, you will need export the results to a file. Please see the Tips & Tricks section for more on exporting query results.

SQL Syntax

Select Records

```
select ID,  
       academic_period,  
       major,  
       major_desc,  
       degree,  
       degree_desc,  
       student_level,  
       student_classification,  
       campus  
from student.student;
```

Use the asterisk to select ALL fields (columns)

```
select *  
from student.student;
```

Note: in MySQL, you must name the database in the from statement – database.table

Select Distinct Records

```
select distinct academic_period  
from student.student;
```

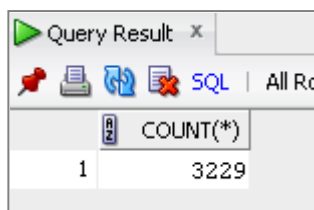
Count Records

When validating data, it can be handy to quickly count the number of records returned in a query. This can be accomplished by one of two methods:

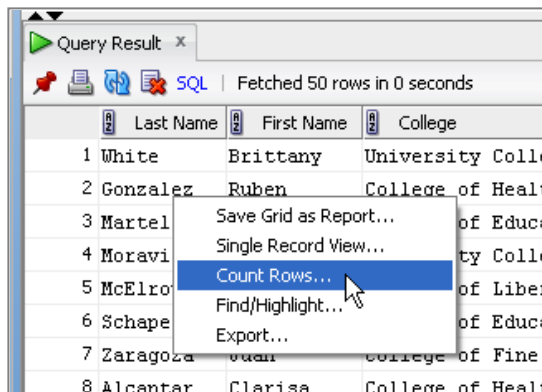
- SQL Syntax Method
 - In the SELECT statement, instead of listing variables, type

SELECT COUNT (*)

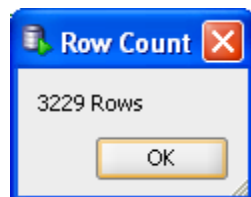
- This counts all the rows in the query and displays the count in the Query Results window



- Query Results Method
 - After a query is run, right-click within the Query Results window
 - Select Count Rows



- The number of rows will be returned in a pop-up window



Where Clause

```
select *
from student.degrees
where academic_period = 201210;
```

Using Like in the Where Clause

```
select *
from student.person
where last_name like 'X%';
```

Joins

- **Inner Join**

```
select *
from student.degrees
join student.person
on degrees.id = person.id
where degrees.academic_period;
```

- **Outer Join**

```
select *
from student.student
left join student.degrees
on student.id = degrees.id
where student.academic_period = 200910;
```

Table/View Aliases

Aliases for tables and views are most convenient when working with multiple tables. Using aliases allows for few keystrokes and cleaner code. For example, the code below does not use aliases:

```
select *
from student.student
left join student.degrees
    on student.id = degrees.id
where student.academic_period = 200910;
```

Using aliases helps make the code more readable and takes less time to type.

```
select *
from student.student s
left join student.degrees d
    on s.id = d.id
where s.academic_period = 200910;
```

Limit cases

```
select *
from student.person
limit 10;
```

Order By

```
select *
from student.person
order by last_name asc;
```

Group By

```
select academic_period, count(id)
from student.student
group by academic_period;
```

Insert Records

```
insert into student.email
(id,internet_address_status,internet_address_type,
internet_address) values
(482915,'Active','PERS','me@hello.com');
```

Delete Records

```
delete from student.email
where id = 482915;
```

Update Records

```
update student.email set internet_address =
'you@hello.com'
where id = 482915;
```

Having

```
select academic_period, count(id)
from student.student
group by academic_period
having count(id) > 10000;
```

Aggregate Functions

```
select academic_period, sum(course_credits)
from student.registrations
where academic_period = 201010;
```

Case

```
select case academic_period
        when 201110 then 'Fall'
        when 201130 then 'Spring'
        when 201150 then 'Summer'
        else 'Error'
        end as term
from student.student
where academic_period in(201110, 201130, 201150);
```

Dates

```
select *
from student.person
where birth_date > '1990-01-01';
```

Subqueries

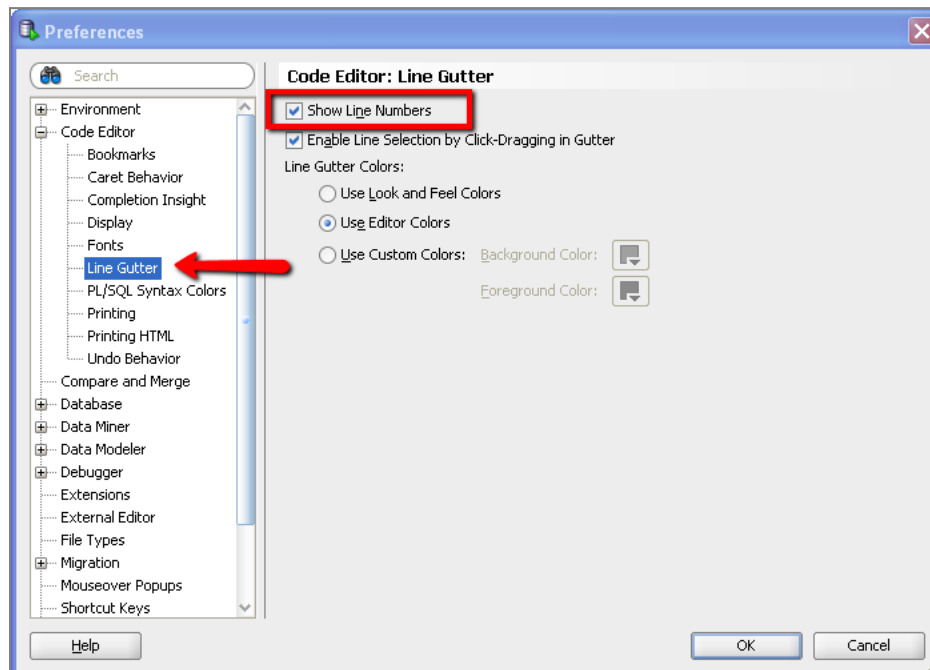
```
select d.*,
       (select max(academic_period)
        from student.registrations r
        where r.id = d.id)
from student.degrees d
where d.academic_period = '201310';
```

SQL Developer Tips & Tricks

Line Numbers

To add line numbers to a SQL Worksheet:

- Go to **Tools > Preferences**
- Expand **Code Editor**
- Select **Line Gutter**
- Check **Show Line Numbers**
- Click **OK**



Completion Insight

Completion Insight gives you suggestions to auto-complete the SQL code you are typing.

- Go to **File > Preferences**
- Expand **Code Editor**
- Select **Completion Insight**
- To turn **off** Completion Insight, uncheck **Enable Completion Auto-Popup in SQL Worksheet**

- To **adjust the speed** at which suggestions appear, move the slider for **Popup Speed**
- Click **OK**

Syntax Colors

You can customize the syntax color and highlight options.

- Go to **File > Preferences**
- Expand **Code Editor**
- Select **PI/SQL Syntax Colors**
- Make desired changes
- Click **OK**

Change Case

If you would prefer a certain case to be maintained in your SQL code, you can define the case rules.

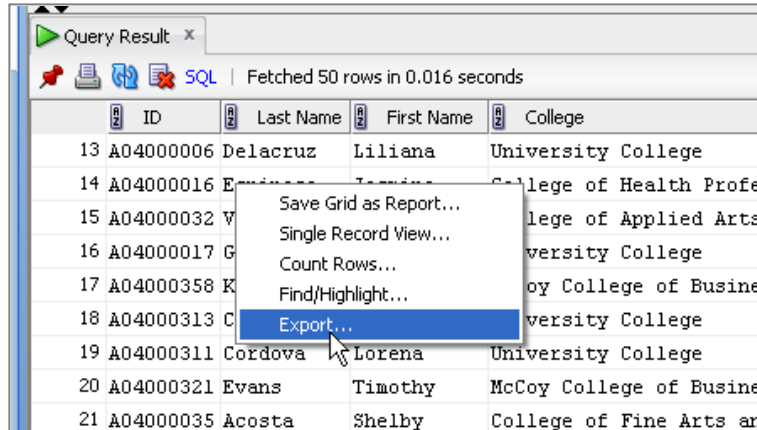
- Go to **File > Preferences**
- Expand **Database**
- Expand **SQL Formatter**
- Select **Oracle Formatting**
- Click **Edit**
- Make desired changes
For example, I prefer all caps when coding
 - Expand **Other**
 - Select **Case change**
 - Select **Whole SQL Uppercase**
 - Click **OK**
 - Click **OK**

Export Query Results

The data you query using SQL Developer can be exported in a number of different formats to allow for further data manipulation.

To export Query Results data:

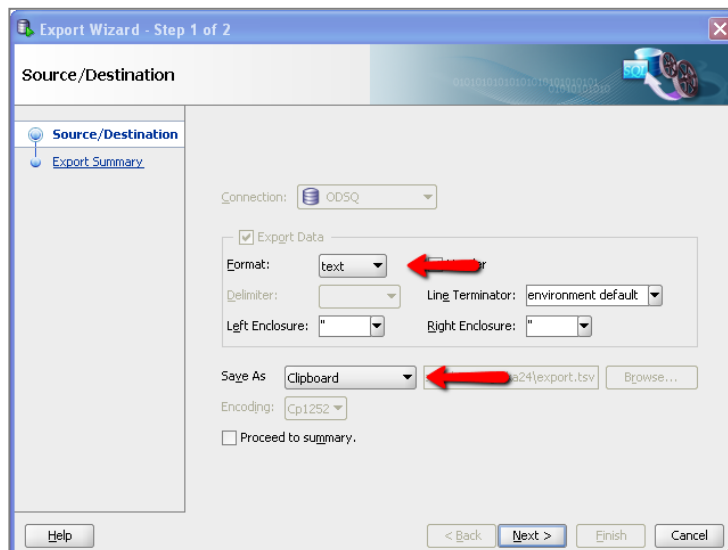
- Right-click anywhere in the **Query Results** window



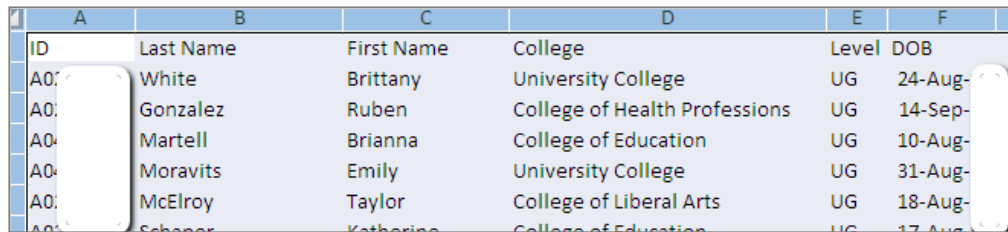
- Select **Export**
- In the **Format** drop-down, you will be given several options as to how you would like to export your data.

A quick way to get data into Excel is to copy the data to the clipboard

- In the **Export Wizard** window, select **text** for the format
- In the Save As drop-down menu, select **Clipboard**



- Click **Next**
- Click **Finish**
- In **Excel**, select a cell to paste the data
- **Paste** the contents of the clipboard by clicking **Paste** on the Home menu or press **Ctrl+V**



ID	Last Name	First Name	College	Level	DOB
A0:	White	Brittany	University College	UG	24-Aug-
A0:	Gonzalez	Ruben	College of Health Professions	UG	14-Sep-
A0:	Martell	Brianna	College of Education	UG	10-Aug-
A0:	Moravits	Emily	University College	UG	31-Aug-
A0:	McElroy	Taylor	College of Liberal Arts	UG	18-Aug-
A0:	Schaefer	Katherine	College of Education	UG	17-Aug-

- The data will be pasted into the worksheet at the desired location

File List

To easily access your saved SQL query files from within SQL Developer, you can use the Files view in the left-panel.

- Go to **View > Files**
- A **tree-view** of your files appears in the left-panel. You can then **navigate** to the location of your saved SQL queries.

To go back to the Connections view, click the **Connections tab** in the left-panel