

PowerSchool Customization

#4 - SQL and Using sqlReports



<https://goo.gl/xyKDXT>

Link (or scan the code) to get a Google copy of this presentation.

Also posted at psugcal.org
(via the "Customization" link)

PowerSchool Customization #4 - SQL and Using sqlReports



CREDITS



thanks to Adam Larsen of
Aurora Educational Technology
for most of the content

aurora
EDUCATIONAL TECHNOLOGY

PLUGINS ▾

CUSTOM ALERTS

NAVIGAT ▾

SERVICES

NEWS

ABOUT

CONTACT

SIGN IN



auroraedtech.com

Aurora Educational Technology has helped hundreds of PowerSchool educational institutions make the most of their student data. Through free plugins, add-ons available for purchase, and individualized customization work, AET has tweaked, enhanced, and improved most areas of the PowerSchool portal.

Tools to develop SQL queries



- Queries can take time to write and troubleshoot.
- A tool to directly query the database is important before you embed your SQL query in its final location.

Oracle SQL Developer

- Two downloads
 - Oracle SQL Developer
 - Java Development Kit (JDK)
- Available from Oracle for free at:
<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>
- Requires internal or VPN connection to the database, can be problematic if hosted



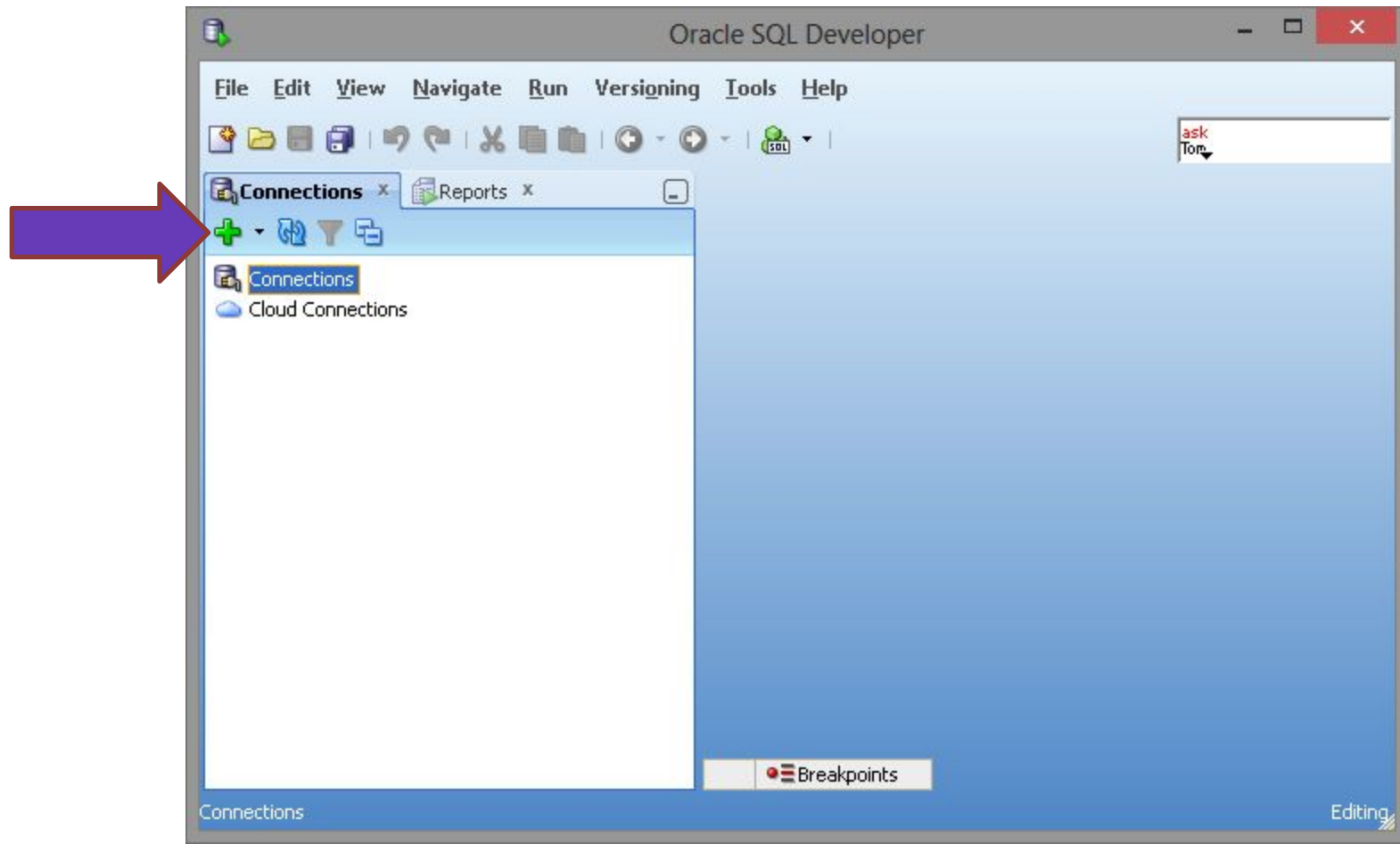
Oracle SQL Developer

Installation instructions

- Run Java Development Kit (JDK) Installer
 - Will also install Java Runtime Environment (JRE)
 - Install JDK into any location
 - Recommended: C:\Program Files\jdk[version number]
- Unpack Oracle SQL Developer into any location
 - Recommended: C:\Program Files\Oracle SQL Developer
- Run sqldeveloper.exe
- Program will prompt you for location of JDK
 - Navigate to location you specified when installing JDK

Oracle SQL Developer

Configuration instructions



Oracle SQL Developer

— — —

New / Select Database Connection

Connection Name Connection Details

Connection Name

Username

Password

☐ Save Password

Oracle Access

Connection Type: Basic Role: default

Hostname: localhost

Port: 1521

☒ SID: xe

☐ Service name

☐ OS Authentication ☐ Kerberos Authentication ☐ Proxy Connection

Status :

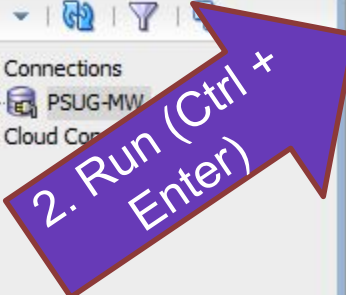
Help Save Clear Test Connect Cancel

Oracle SQL Developer

The screenshot displays the 'New / Select Database Connection' dialog box in Oracle SQL Developer. The dialog is divided into two main sections: 'Connection Name' and 'Connection Details'. The 'Connection Name' section contains a table with one row showing 'PSUG'. The 'Connection Details' section contains several fields and options:

- Connection Name:** PSUG
- Username:** psnavigator
- Password:** (empty)
- ☐ Save Password
- Oracle Access:**
 - Connection Type:** Basic
 - Role:** default
 - Hostname:** (empty)
 - Port:** 1521
 - ☒ SID: psproddb
 - ☐ Service name: (empty)
 - ☐ OS Authentication
 - ☐ Kerberos Authentication
 - ☐ Proxy Connection

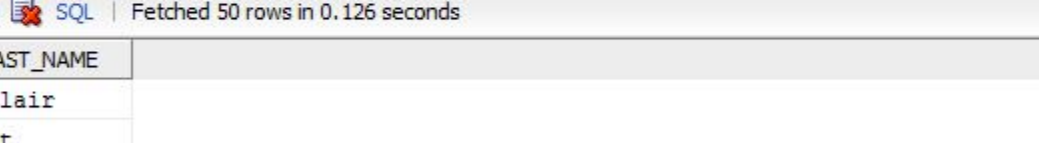
At the bottom of the dialog, there is a 'Status :' label and a row of buttons: Help, Save, Clear, Test, and Connect. A large blue arrow points to the 'Connect' button.



1. Query



3. Result



Query Result x

SQL | Fetched 50 rows in 0.126 seconds

	LAST_NAME
1	Auclair
2	Aust
3	Avery
4	Babb
5	Bailey
6	Baker
7	Barlow
8	Barlow
9	Bartlett
10	Baumann
11	Bennett

3. Result

SQL Studio



Data Dictionary

students

Select a field

+ Insert

↻ Update

Students ×

new 1 ×



▶ Run

⚙ Settings

? Help

Querystring:

Ex: schoolid=1&frn=001969

↗ sqlReport

↗ tlist_sql

↗ PowerQuery

1 SELECT

2 students.lastfirst,

3 students.gra|

students.grade_level PS

students.gradreqset PS

students.gradreqsetid PS

students.graduated_rank PS

students.graduated_schoolid PS

students.graduated_schoolname PS

sg_students.gradelevel PS

studenttest.grade_level PS

- Soon to be released!
- Embedded in PowerSchool

SQL Studio



- SOON TO BE RELEASED at **auroraedtech.com**
- Web-based GUI for developing SQL queries
- No VPN needed
- Works with hosted customers
- Access to entire data dictionary, including views
- Autocomplete tables and fields
- Add querystring variables and PSHTML
- Export to sqlReports template, tlist_sql or PowerQuery
- Library of sql queries
- Built-in Data Dictionary

auroraedtech.com



Test Server

URL: `pstest.psugcal.org/admin`

User: `XXaghs1`

Pwd: `aghs1`

XX= number between 01 and 99

(example: `25aghs1`)

Functions

[Attendance](#)
[Daily Bulletin](#)
[Enrollment Summary](#)
[Master Schedule](#)
[Dashboard](#)
[Special Functions](#)
[Teacher Schedules](#)

Reports

[System Reports](#)
[ReportWorks](#)

People

[Student Search](#)
[Staff Search](#)
[Contact Search](#)
[Enroll New Student](#)
[New Staff Entry](#)
[New Contact Entry](#)

Setup

[School](#)
[System](#)
[Personalize](#)

SQL Studio

Data Dictionary

Select a table

Select a table

+ Insert

Update

new 1



Run

Settings

Help

Querystring:

Ex: schoolid=1&frn=001969

tlist_sql

PowerQuery

```

1 SELECT
2 name
3
4 FROM
5 schools
    
```

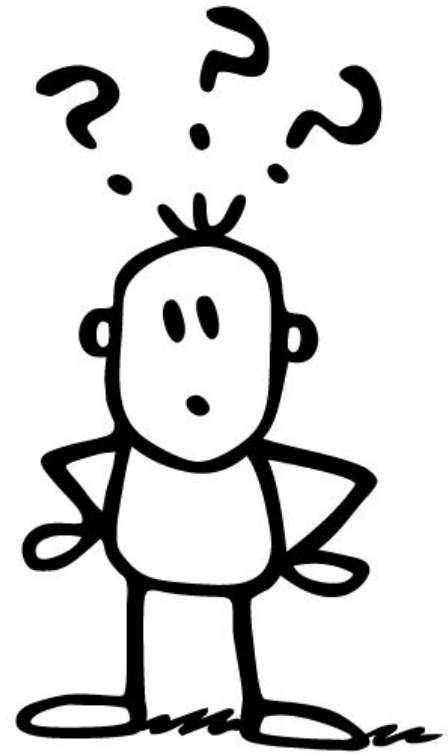
Returned 30 row(s)

Copy Data

name
Graduated Students
Apple Grove High School 3
Apple Grove High School
Apple Grove High School 2
Scheduling High School 1
Cherry Hill Middle School
Washington Elementary



What is SQL?



What is SQL?

- **Structured Query Language**
- ANSI-standard language used to interact with relational databases
 - Still comes in a few flavors (MySQL, **Oracle**, MSSQL)
- Performs four chief functions (CRUD):
 - Create
 - Read
 - Update
 - Delete

SQL and PowerSchool

- Oracle flavor of SQL
- SQL is *the fastest* way to extract data from PowerSchool
- Can only read from the Oracle database
 - Create, update, and delete violate the Terms of Service with Pearson
 - Support might not be able to help you when you mess something up

Getting Started



How do I get started?

- psnavigator Oracle account
 - Configured in System
- Software options
 - Oracle SQL Developer
 - oss.oracle.com/sqldeveloper.html
 - Good
 - Free
 - Can schedule jobs
 - No additional drivers needed
 - Bad
 - Requires JAVA
 - Only connects to Oracle

How do I get started?

- Software options (cont.)
 - Navicat
 - Navicat.com (PremiumSoft)
 - Good
 - Free lite versions available via search
 - Can schedule jobs
 - Connects to all major databases
 - Very lightweight program - runs quickly
 - Bad
 - Need to install Oracle OCI Drivers
 - Non-commercial Premium edition is \$599 USD
 - Non-commercial Oracle-only edition is \$299 USD
 - Only the paid version can schedule jobs

Available from
auroraedtech.com

How do I get started?

- Software options (cont.)
 - SQL Server Reporting Services
 - Microsoft.com
 - Good
 - You may already license it
 - Feature-rich reporting suite
 - Can schedule jobs
 - Bad
 - Requires Microsoft overhead
 - Building reports can be complicated
 - Takes some fiddling to work with Oracle

How do I get started?

- Inside PowerSchool
 - sqlReports
 - sqlreports.net (Matt Freund)
 - Good
 - Embeds right into PowerSchool
 - Easy for end-users to run your queries at will
 - Can take user input when executed
 - Scheduled jobs
 - Bad
 - Not a development environment
 - Requires workarounds for queries that contain certain characters

How do I get started?

- Inside PowerSchool
 - Enterprise Reporting (APEX)
 - <https://support.powerschool.com/article/77492>
 - [psugcal.org/index.php?title=Enterprise Reporting](https://psugcal.org/index.php?title=Enterprise%20Reporting)
 - Good
 - Part of the product supported by PowerSchool
 - Easy for end-users to run your queries at will
 - Many options to use reports flexibly for different needs, including basic charting
 - Bad
 - Not a development environment
 - Not an easy development interface
 - No scheduled or emailed reports

How do I get started?

- Inside PowerSchool
 - PowerQueries
 - <https://support.powerschool.com/developer>
 - psugcal.org/index.php?title=PowerQueries
 - Good
 - Part of the product supported by PowerSchool
 - Available in Database Export Manager
 - Can be scheduled
 - Can be used by customizations and partners both internally and externally
 - Bad
 - Difficult syntax, not easy to create
 - Must be installed as a plugin

How do I get started?

- Inside PowerSchool
 - In customizations using `tlist_sql`
 - More on this later
 - Good
 - Can be integrated directly in PS
 - Powerful customizations
 - Bad
 - Requires advanced customization skill
 - Deprecated by PowerSchool
 - Issues with security and "layering"

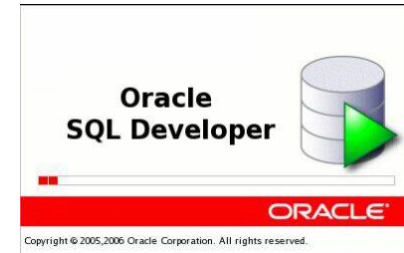
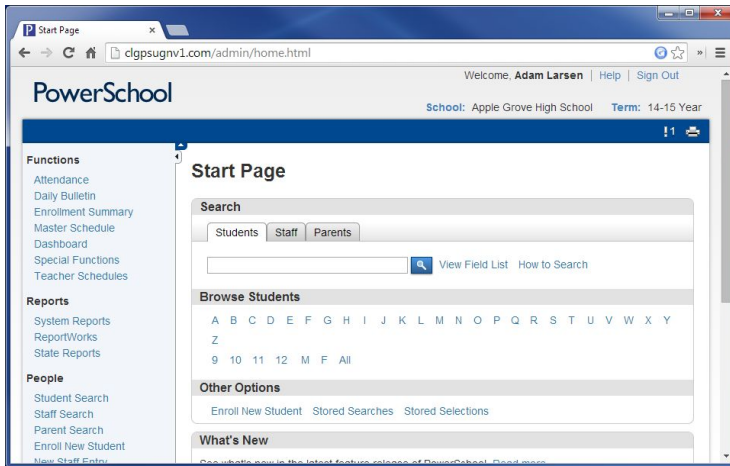
How do I get started?



- PowerSchool Data Dictionaries
 - <https://support.powerschool.com/dir/5933>
 - (Start with Data Dictionary Tables)
 - As of 12.1.1 they are online via the Help menu
- Tech on the Net
 - <http://www.techonthenet.com/sql/index.php>
- W3 Schools
 - <http://www.w3schools.com/sql>

Database Intro

Where is the database?



Microsoft®
SQL Server®
Reporting Services

sqlReports Overview

Since 2009, PowerSchool users have been able to easily call sqlReports. Dean Dahlvang released a free PowerSchool reporting add-on that can turn simple to medium sql queries anywhere in the database, and it became one of the most popular releases, with sqlReports 3 being the last on



Relational Databases

- **Are sort of like an Excel spreadsheet**

- Store data in tables
- Have different data types
- Can be searched

- **Are not at all like an Excel spreadsheet**

- Tables only store the data they need
- Unique identifiers link tables together
- The results of a query may come from several tables at once

SELECT Statement

SELECT

somestuff

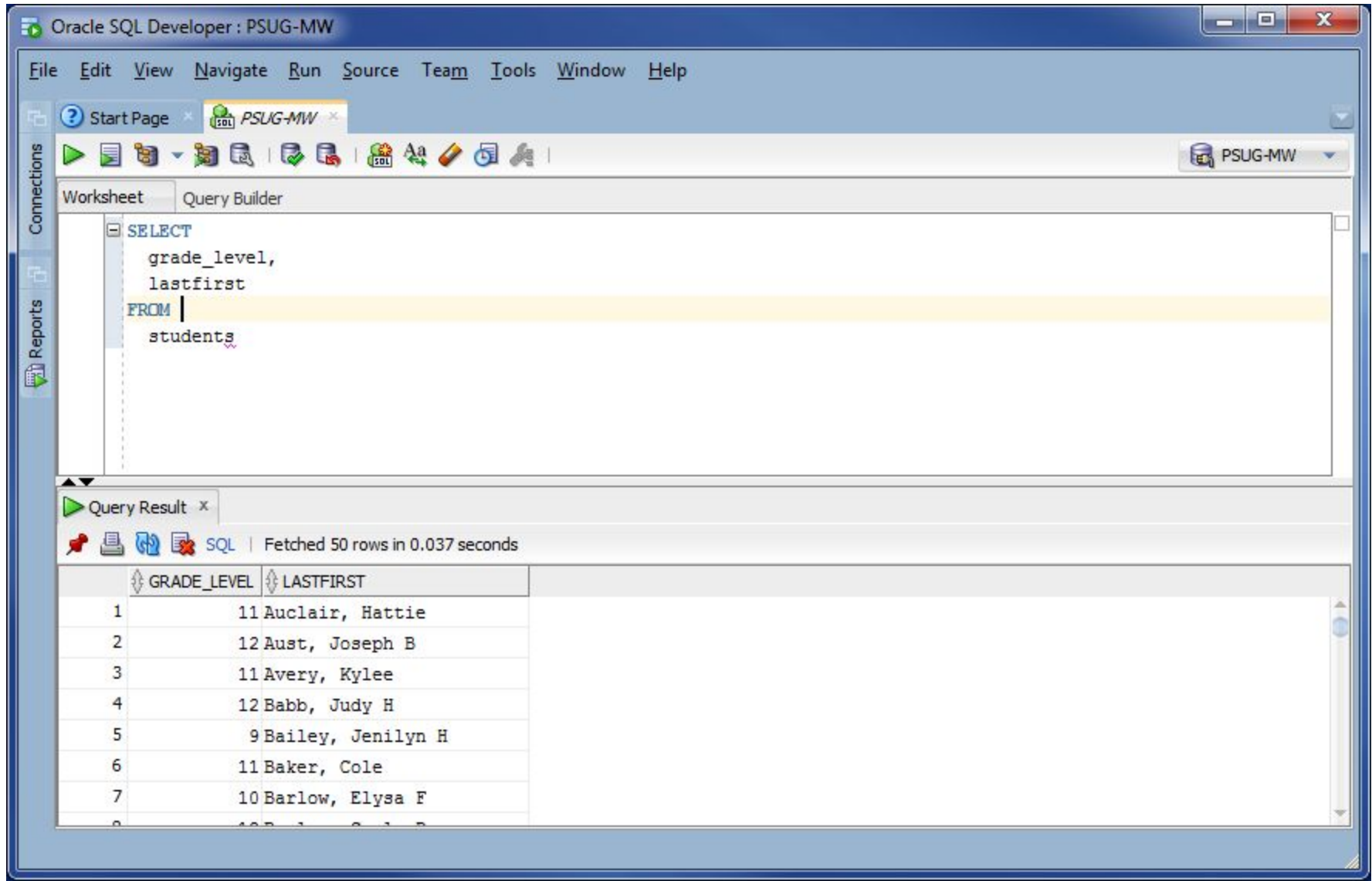
FROM

somewhere

SELECT statement

- All of our queries will begin with SELECT
- Required pieces
 - SELECT keyword
 - List of column names
 - FROM keyword
 - Table name

SELECT statement (cont.)



The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL script in the 'Worksheet' tab with the following query:

```
SELECT
  grade_level,
  lastfirst
FROM
  students
```

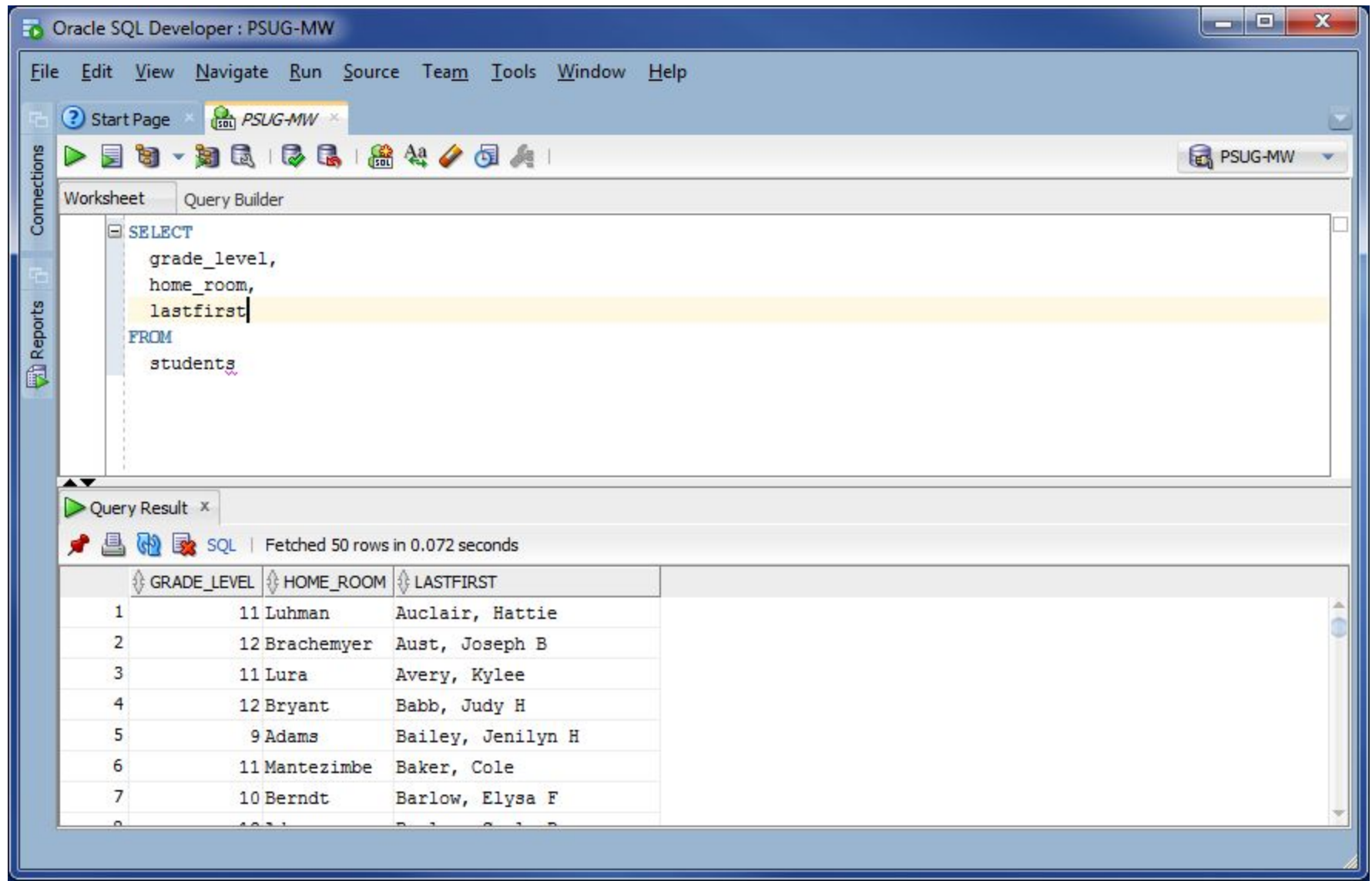
Below the script, the 'Query Result' tab is active, showing the results of the query. The results are displayed in a table with two columns: 'GRADE_LEVEL' and 'LASTFIRST'. The table contains 10 rows of data, with the first 7 rows visible in the screenshot.

	GRADE_LEVEL	LASTFIRST
1	11	Auclair, Hattie
2	12	Aust, Joseph B
3	11	Avery, Kylee
4	12	Babb, Judy H
5	9	Bailey, Jenilyn H
6	11	Baker, Cole
7	10	Barlow, Elysa F
8	10	Barlow, Elysa F
9	10	Barlow, Elysa F
10	10	Barlow, Elysa F

Formatting

- CAPITALIZATION does not matter
 - Some prefer to type keywords as ALL CAPS
 - Be careful on comparisons: 'aghs' != 'AGHS'
- Spaces between keywords are required
- Tabbing and new lines are optional but recommended
 - `SELECT field1, field2 FROM tables`
 - `OR`
 - `SELECT`
 - `field1,`
 - `field2`
 - `FROM`
 - `tables`

SELECT statement (cont.)

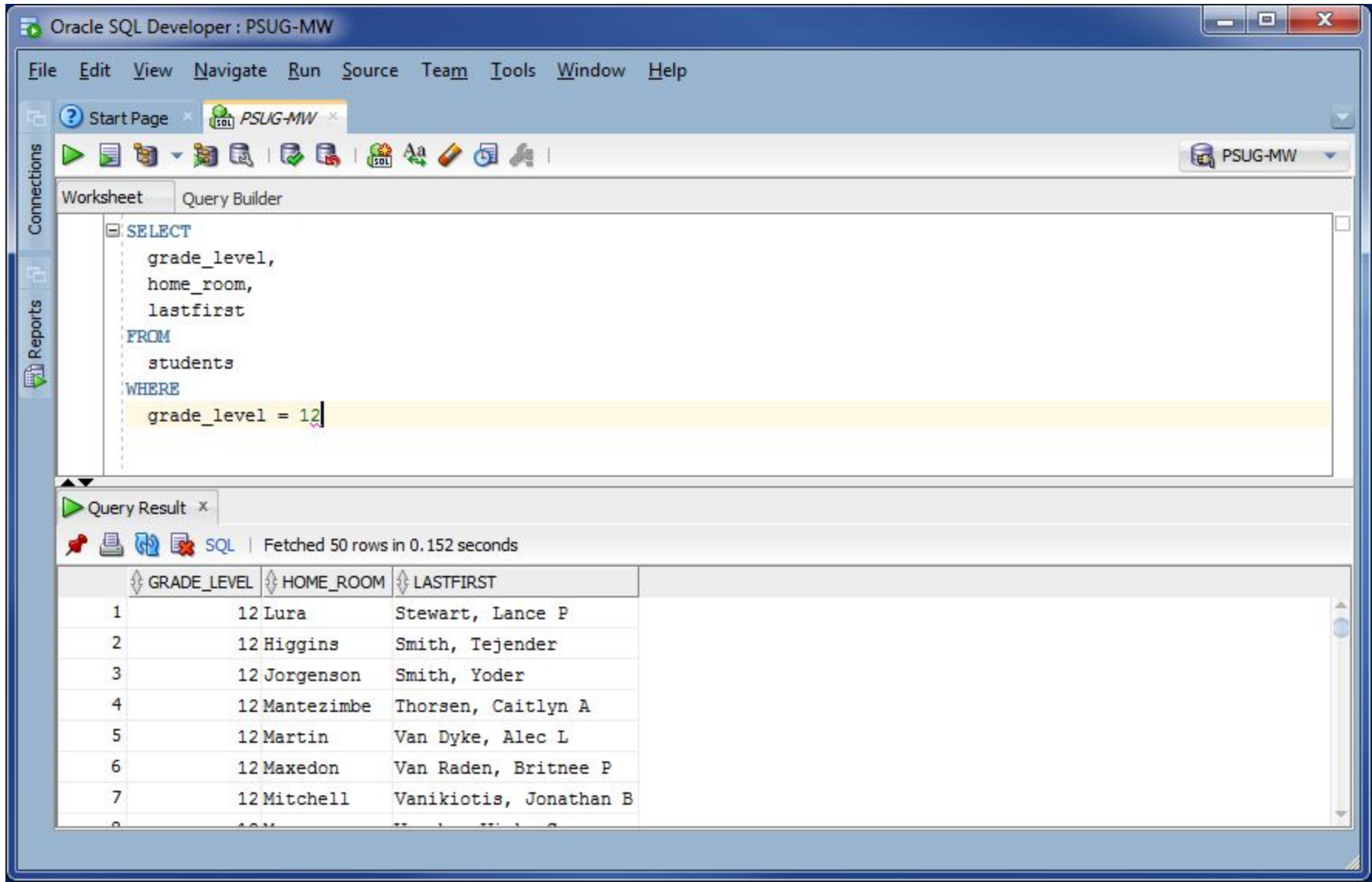


The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL script in the 'Worksheet' tab. The script is a SELECT statement that retrieves the 'grade_level', 'home_room', and 'lastfirst' columns from the 'students' table. The 'lastfirst' column is highlighted in yellow. Below the script, the 'Query Result' tab is active, showing the results of the query. The results are displayed in a table with three columns: 'GRADE_LEVEL', 'HOME_ROOM', and 'LASTFIRST'. The table contains 8 rows of data, with the first 7 rows visible. The status bar indicates that 50 rows were fetched in 0.072 seconds.

```
SELECT
  grade_level,
  home_room,
  lastfirst
FROM
  students
```

	GRADE_LEVEL	HOME_ROOM	LASTFIRST
1	11	Luhman	Auclair, Hattie
2	12	Brachemyer	Aust, Joseph B
3	11	Lura	Avery, Kylee
4	12	Bryant	Babb, Judy H
5	9	Adams	Bailey, Jenilyn H
6	11	Mantezimbe	Baker, Cole
7	10	Berndt	Barlow, Elysa F
8	10	Berndt	Barlow, Elysa F

WHERE clause and = operator



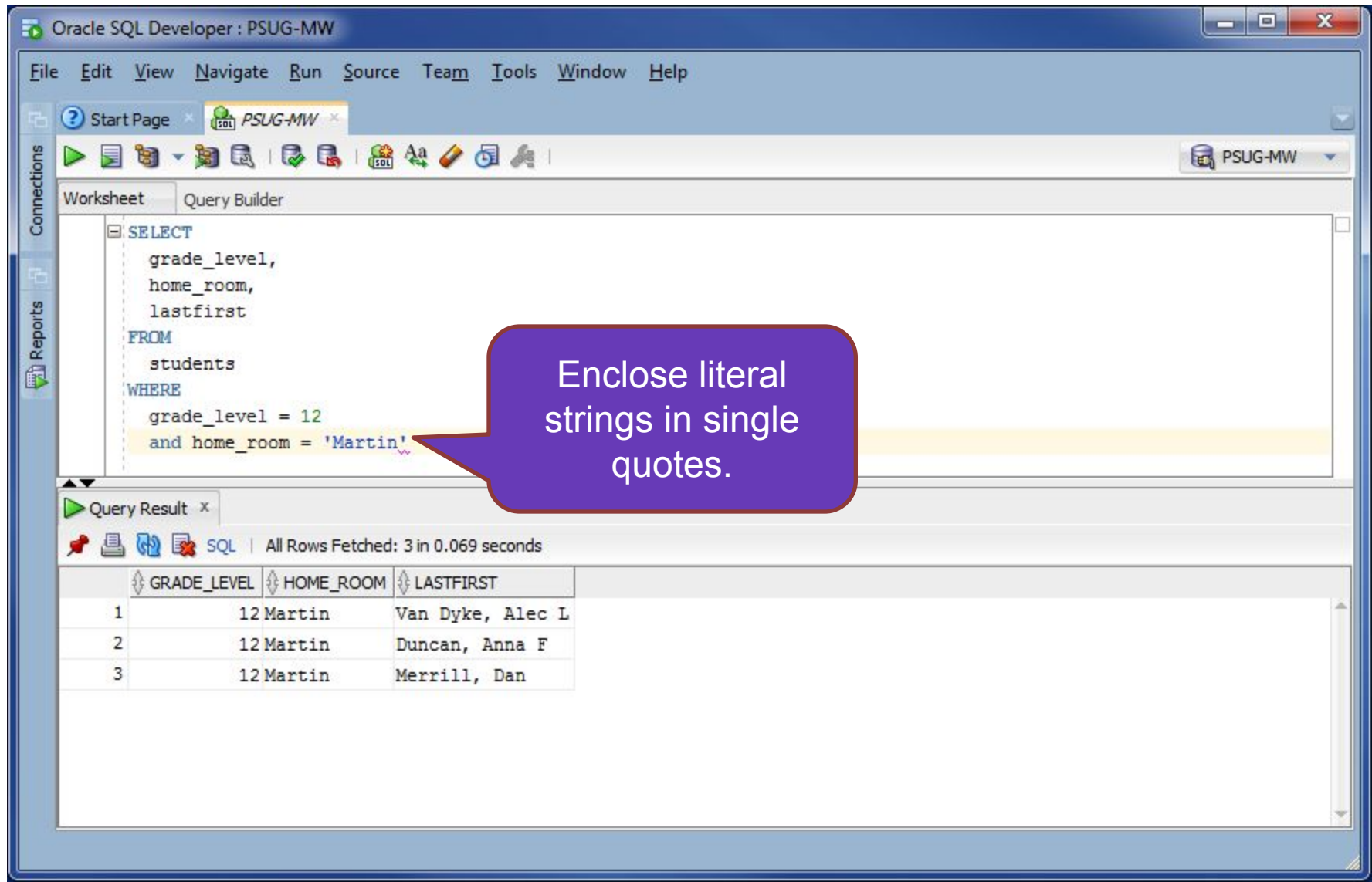
The screenshot shows the Oracle SQL Developer interface with a query window titled 'PSUG-MW'. The query is as follows:

```
SELECT
  grade_level,
  home_room,
  lastfirst
FROM
  students
WHERE
  grade_level = 12
```

The 'Query Result' window below shows the results of the query, fetched in 0.152 seconds. The results are displayed in a table with columns: GRADE_LEVEL, HOME_ROOM, and LASTFIRST.

	GRADE_LEVEL	HOME_ROOM	LASTFIRST
1	12	Lura	Stewart, Lance P
2	12	Higgins	Smith, Tejender
3	12	Jorgenson	Smith, Yoder
4	12	Mantezimbe	Thorsen, Caitlyn A
5	12	Martin	Van Dyke, Alec L
6	12	Maxedon	Van Raden, Britnee P
7	12	Mitchell	Vanikiotis, Jonathan B

AND operator



The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab:

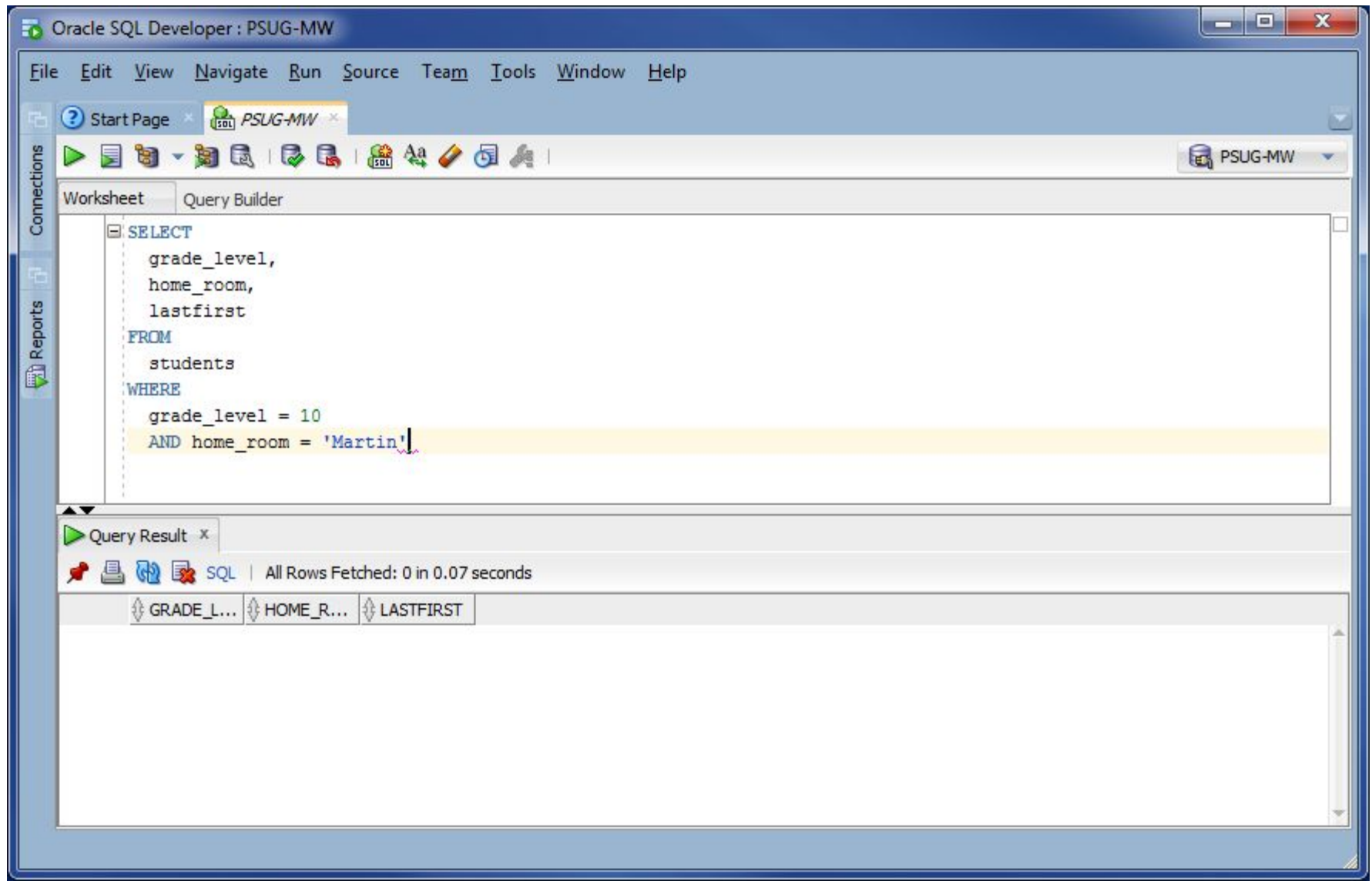
```
SELECT  
  grade_level,  
  home_room,  
  lastfirst  
FROM  
  students  
WHERE  
  grade_level = 12  
  and home_room = 'Martin'
```

A purple callout bubble points to the string 'Martin' in the WHERE clause, containing the text: "Enclose literal strings in single quotes."

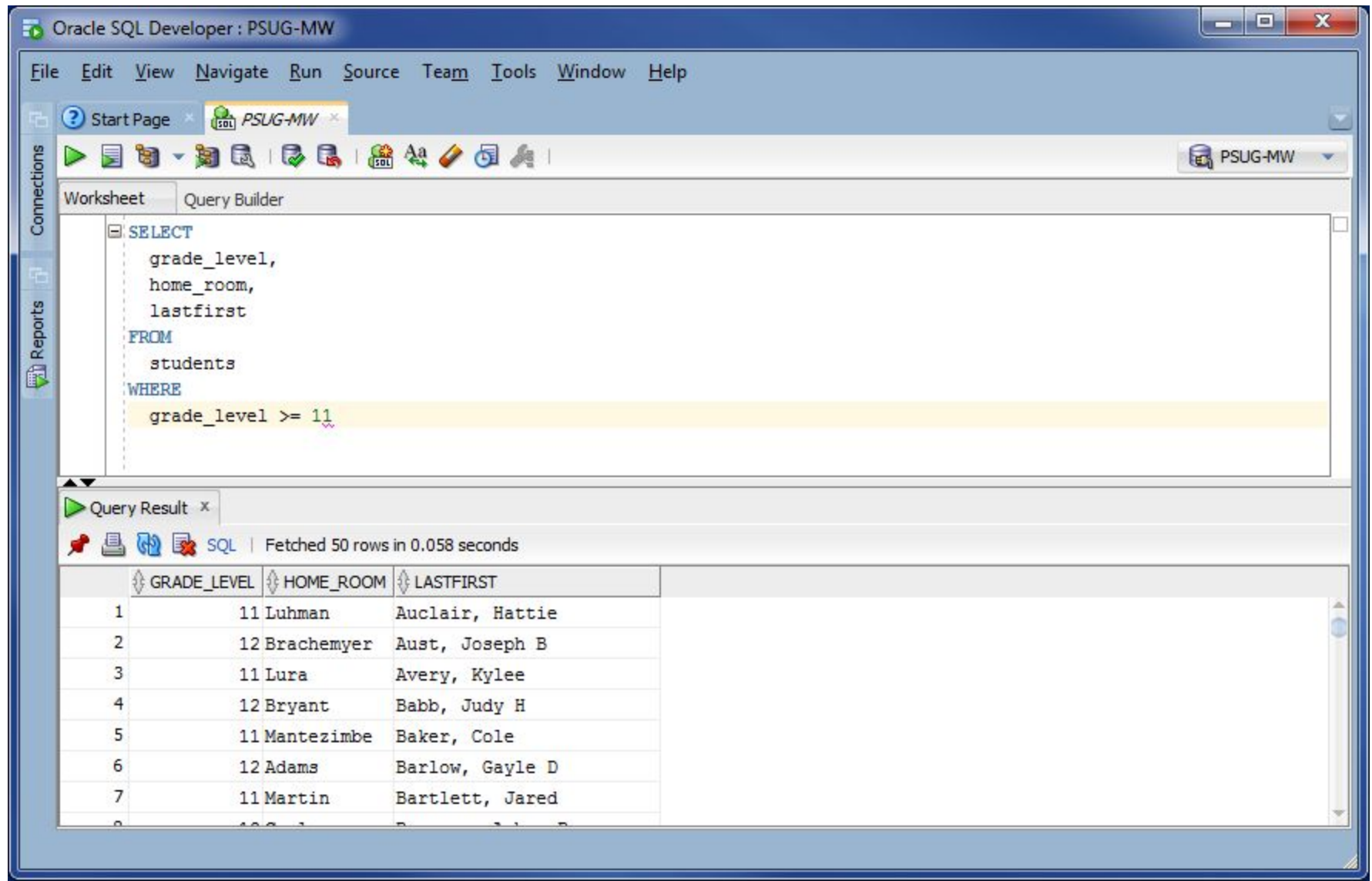
Below the query, the Query Result tab shows the results of the query. The status bar indicates "All Rows Fetched: 3 in 0.069 seconds".

	GRADE_LEVEL	HOME_ROOM	LASTFIRST
1	12	Martin	Van Dyke, Alec L
2	12	Martin	Duncan, Anna F
3	12	Martin	Merrill, Dan

AND operator



>= operator



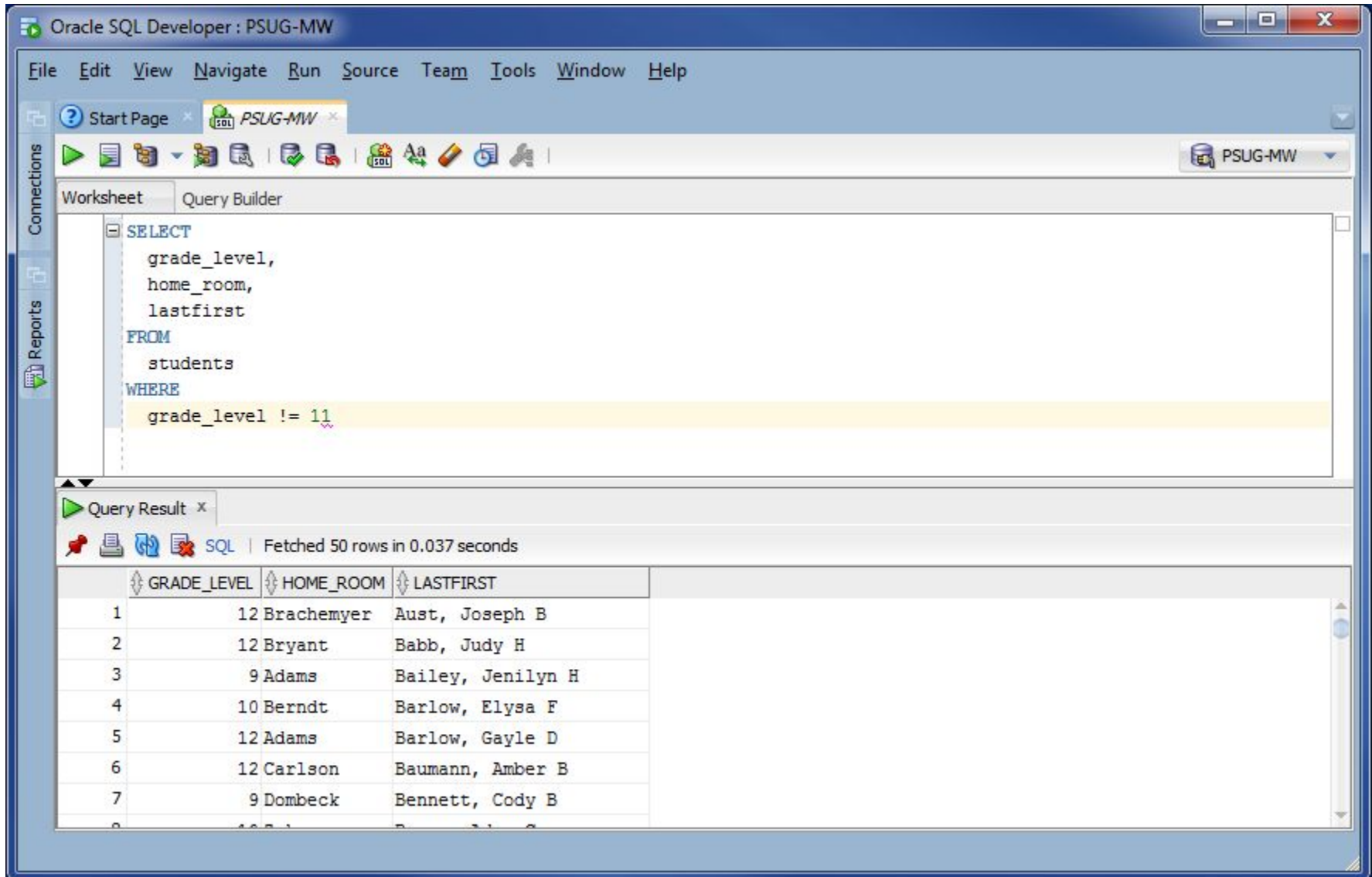
The screenshot shows the Oracle SQL Developer interface with a query window titled 'PSUG-MW'. The query is as follows:

```
SELECT
  grade_level,
  home_room,
  lastfirst
FROM
  students
WHERE
  grade_level >= 11
```

The 'Query Result' window below shows the results of the query, fetched in 0.058 seconds. The results are displayed in a table with columns: GRADE_LEVEL, HOME_ROOM, and LASTFIRST.

	GRADE_LEVEL	HOME_ROOM	LASTFIRST
1	11	Luhman	Auclair, Hattie
2	12	Brachemyer	Aust, Joseph B
3	11	Lura	Avery, Kylee
4	12	Bryant	Babb, Judy H
5	11	Mantezimbe	Baker, Cole
6	12	Adams	Barlow, Gayle D
7	11	Martin	Bartlett, Jared

!= operator



The screenshot shows the Oracle SQL Developer interface with a query window titled "PSUG-MW". The query is as follows:

```
SELECT
  grade_level,
  home_room,
  lastfirst
FROM
  students
WHERE
  grade_level != 11
```

The query result is displayed in a table with the following columns: GRADE_LEVEL, HOME_ROOM, and LASTFIRST. The table contains 8 rows of data.

	GRADE_LEVEL	HOME_ROOM	LASTFIRST
1	12	Brachemyer	Aust, Joseph B
2	12	Bryant	Babb, Judy H
3	9	Adams	Bailey, Jenilyn H
4	10	Berndt	Barlow, Elysa F
5	12	Adams	Barlow, Gayle D
6	12	Carlson	Baumann, Amber B
7	9	Dombeck	Bennett, Cody B
8	10	Adams	Bennett, Cody B

OR operator

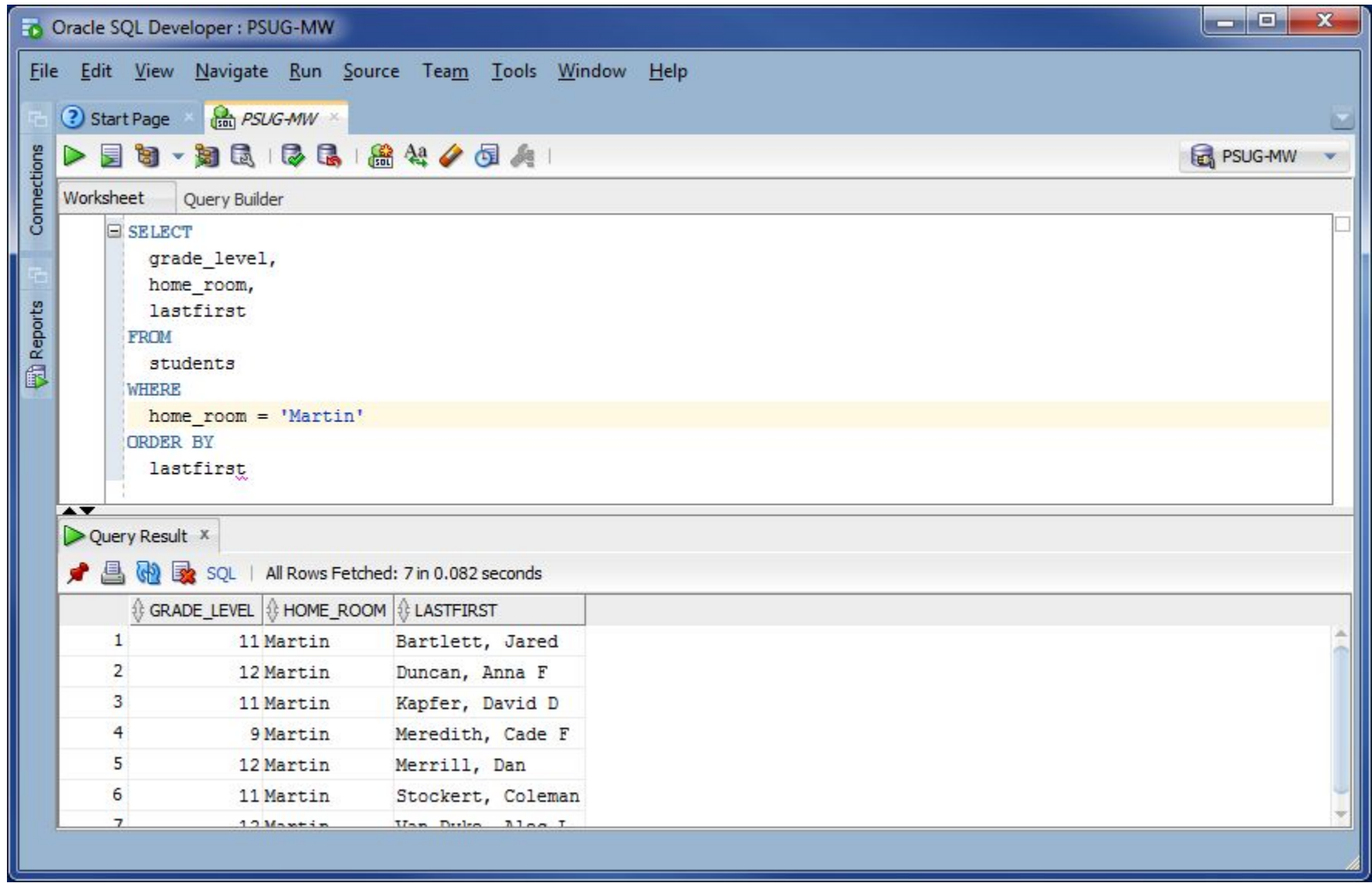
The screenshot shows the Oracle SQL Developer interface with a query window titled 'PSUG-MW'. The query is as follows:

```
SELECT
  grade_level,
  home_room,
  lastfirst
FROM
  students
WHERE
  home_room = 'Martin'
  OR home_room = 'Thatcher'
```

The query results are displayed in the 'Query Result' window, showing 13 rows fetched in 0.085 seconds. The results are as follows:

	GRADE_LEVEL	HOME_ROOM	LASTFIRST
1	11	Martin	Bartlett, Jared
2	12	Martin	Duncan, Anna F
3	11	Thatcher	Croucher, Dillon H
4	12	Martin	Merrill, Dan
5	12	Thatcher	Hansen, Cody P
6	11	Martin	Kapfer, David D
7	9	Martin	Meredith, Cade F
8	11	Thatcher	Nichols, C
9	12	Martin	Roberts, J
10	11	Thatcher	Roberts, J
11	12	Martin	Roberts, J
12	11	Thatcher	Roberts, J
13	12	Martin	Roberts, J

ORDER BY clause



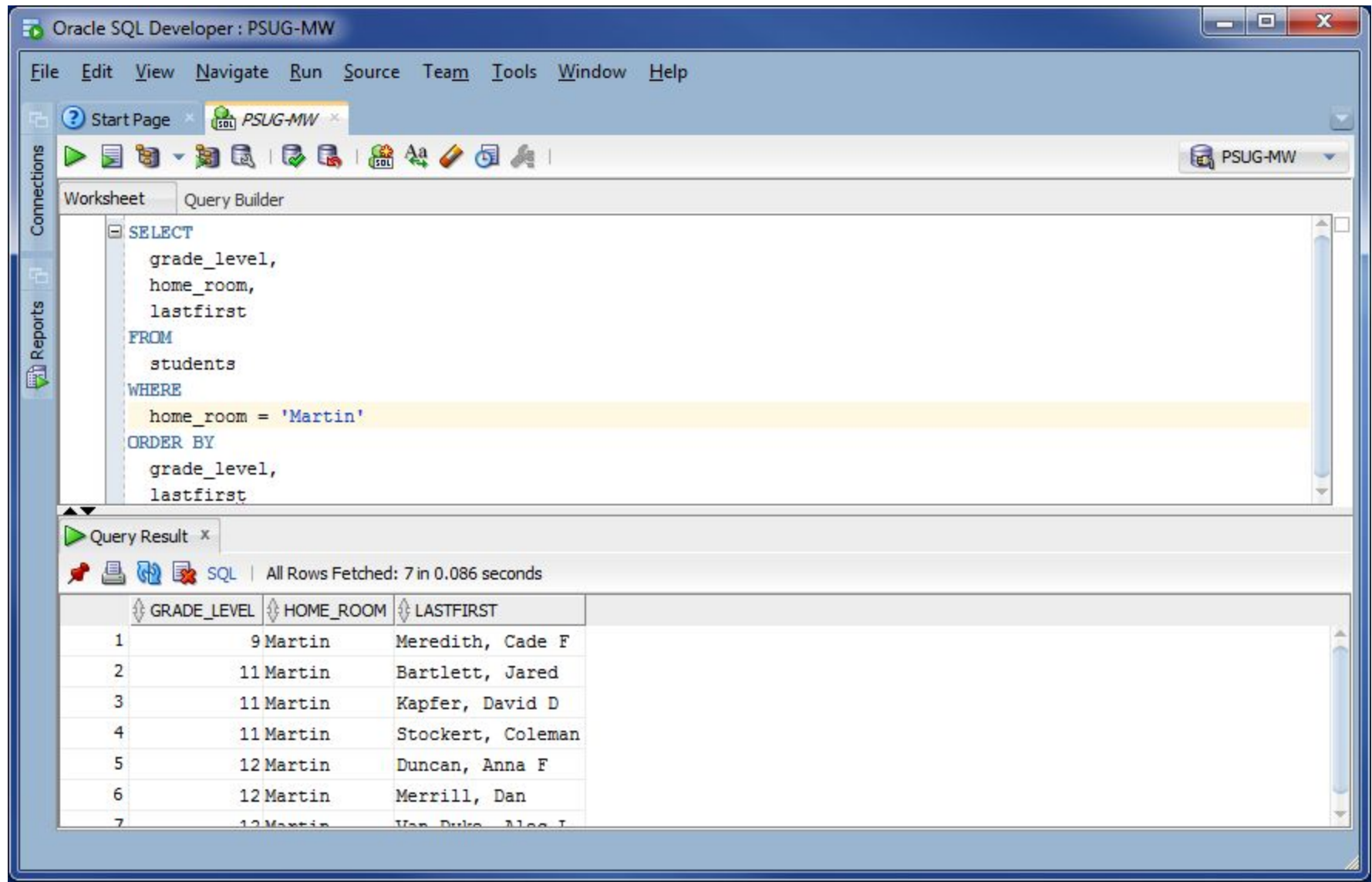
The screenshot shows the Oracle SQL Developer interface with a query window titled 'PSUG-MW'. The query is as follows:

```
SELECT
  grade_level,
  home_room,
  lastfirst
FROM
  students
WHERE
  home_room = 'Martin'
ORDER BY
  lastfirst
```

The 'Query Result' window below shows the results of the query, sorted by the 'lastfirst' column. The results are as follows:

	GRADE_LEVEL	HOME_ROOM	LASTFIRST
1	11	Martin	Bartlett, Jared
2	12	Martin	Duncan, Anna F
3	11	Martin	Kapfer, David D
4	9	Martin	Meredith, Cade F
5	12	Martin	Merrill, Dan
6	11	Martin	Stockert, Coleman
7	12	Martin	Van Duke, Alex J

ORDER BY clause



The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab. The query is as follows:

```
SELECT
  grade_level,
  home_room,
  lastfirst
FROM
  students
WHERE
  home_room = 'Martin'
ORDER BY
  grade_level,
  lastfirst
```

The Query Result tab shows the results of the query, which are 7 rows of data. The columns are GRADE_LEVEL, HOME_ROOM, and LASTFIRST. The data is sorted by grade_level and then by lastfirst.

	GRADE_LEVEL	HOME_ROOM	LASTFIRST
1	9	Martin	Meredith, Cade F
2	11	Martin	Bartlett, Jared
3	11	Martin	Kapfer, David D
4	11	Martin	Stockert, Coleman
5	12	Martin	Duncan, Anna F
6	12	Martin	Merrill, Dan
7	12	Martin	Van Duke, Alec J

ORDER BY clause

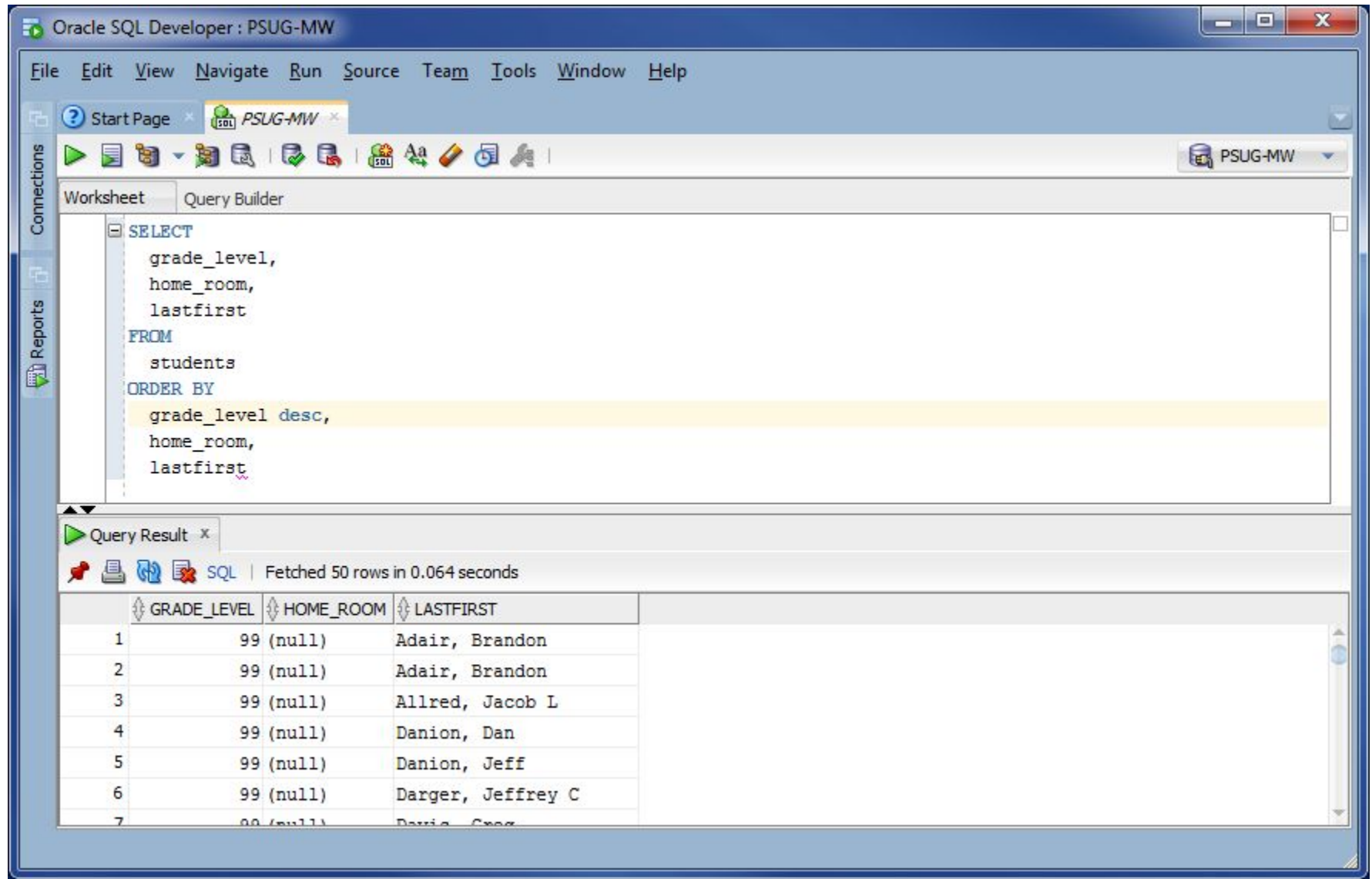
The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab. The query is as follows:

```
SELECT
  grade_level,
  home_room,
  lastfirst
FROM
  students
ORDER BY
  grade_level,
  home_room,
  lastfirst
```

The 'Query Result' tab at the bottom shows the results of the query. It indicates that 50 rows were fetched in 0.068 seconds. The results are displayed in a table with the following columns: GRADE_LEVEL, HOME_ROOM, and LASTFIRST. The first seven rows are visible:

	GRADE_LEVEL	HOME_ROOM	LASTFIRST
1	0	Abbot	Alu, Matthew G
2	0	Abbot	Brush, Jed G
3	0	Abbot	Daniels, Nick
4	0	Abbot	Fredericks, Johnny
5	0	Abbot	Hamilton, Kirsten B
6	0	Abbot	Hennemann, Garrison B
7	0	Abbot	Jones, Steven M

ORDER BY clause, DESC keyword



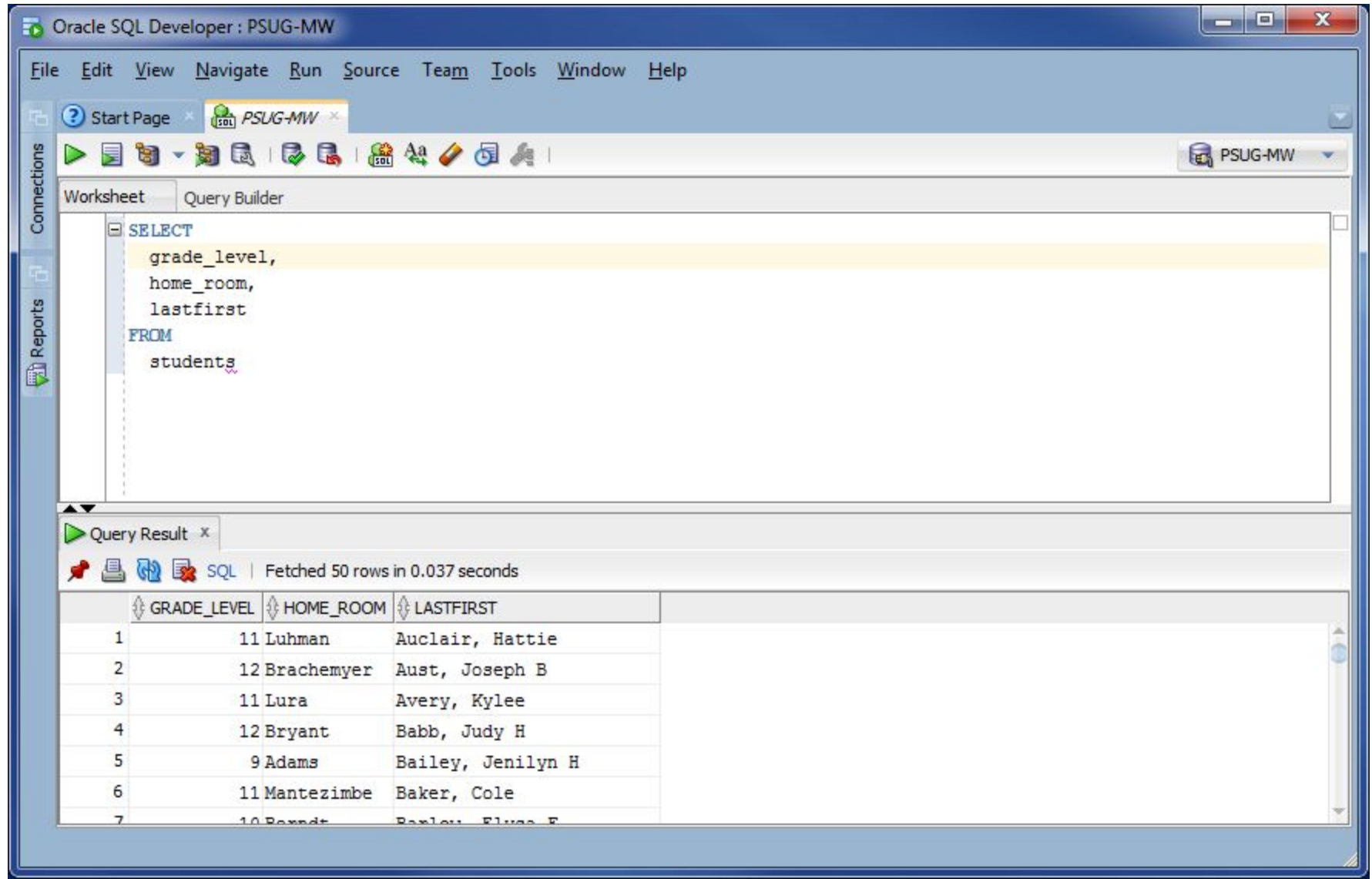
The screenshot shows the Oracle SQL Developer interface. The main window displays an SQL query in the Worksheet tab. The query is as follows:

```
SELECT
  grade_level,
  home_room,
  lastfirst
FROM
  students
ORDER BY
  grade_level desc,
  home_room,
  lastfirst
```

The query has been executed, and the results are displayed in the Query Result tab. The results show 50 rows fetched in 0.064 seconds. The columns are GRADE_LEVEL, HOME_ROOM, and LASTFIRST. The results are ordered by grade_level in descending order, then by home_room, and finally by lastfirst.

	GRADE_LEVEL	HOME_ROOM	LASTFIRST
1	99 (null)		Adair, Brandon
2	99 (null)		Adair, Brandon
3	99 (null)		Allred, Jacob L
4	99 (null)		Danion, Dan
5	99 (null)		Danion, Jeff
6	99 (null)		Darger, Jeffrey C
7	99 (null)		Davis, Greg

SELECT statement (cont.)



The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab:

```
SELECT
  grade_level,
  home_room,
  lastfirst
FROM
  students
```

Below the query, the Query Result tab shows the results of the query. The results are displayed in a table with 4 columns: GRADE_LEVEL, HOME_ROOM, LASTFIRST, and an unnamed column. The table contains 7 rows of data.

	GRADE_LEVEL	HOME_ROOM	LASTFIRST	
1	11	Luhman	Auclair, Hattie	
2	12	Brachemyer	Aust, Joseph B	
3	11	Lura	Avery, Kylee	
4	12	Bryant	Babb, Judy H	
5	9	Adams	Bailey, Jenilyn H	
6	11	Mantezimbe	Baker, Cole	
7	10	Forndt	Barlow, Eliza E	

DISTINCT keyword

The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab:

```
SELECT DISTINCT
  grade_level,
  home_room,
  lastfirst
FROM
  students
```

Below the query, the Query Result tab shows the results of the query. The status bar indicates "Fetched 50 rows in 0.078 seconds". The results are displayed in a table with columns: GRADE_LEVEL, HOME_ROOM, and LASTFIRST.

	GRADE_LEVEL	HOME_ROOM	LASTFIRST
1	11	Lura	Avery, Kylee
2	11	Mitchell	Black, Javier S
3	12	Adams	Bouk, Lauri J
4	12	Bean	Anderson, Trevor B
5	12	Derringer	Brewster, Annie G
6	12	Hastings	Chesnut, Patti L
7	11	Crane	Brickman, Daniel C

DISTINCT keyword

The screenshot shows the Oracle SQL Developer interface with the title bar "Oracle SQL Developer : PSUG-MW". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains icons for Start Page, Run, Save, Undo, Redo, and others. The left sidebar shows the Connections and Reports panels. The main workspace is divided into a Worksheet and a Query Builder. The Worksheet contains the following SQL query:

```
SELECT DISTINCT
  grade_level,
  home_room
FROM
  students
```

The Query Result panel at the bottom shows the results of the query. It indicates that 50 rows were fetched in 0.096 seconds. The results are displayed in a table with two columns: GRADE_LEVEL and HOME_ROOM.

	GRADE_LEVEL	HOME_ROOM
1	0	Abbot
2	0	Adair
3	0	Allen
4	0	Arnold
5	0	(null)
6	1	Barker
7	1	Blain

DISTINCT keyword

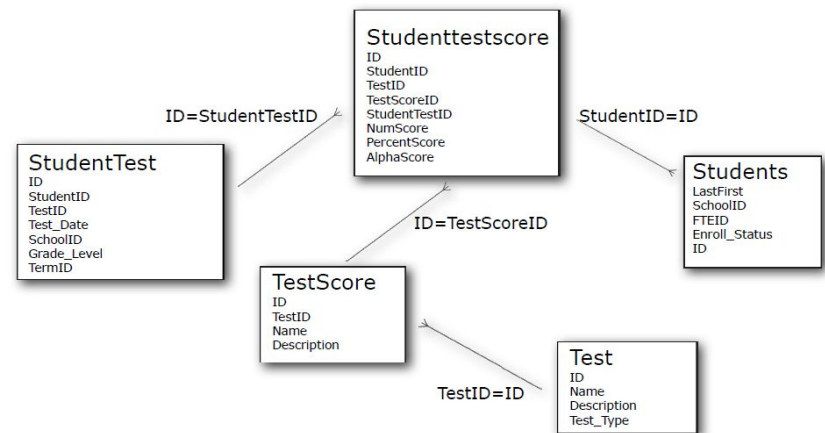
The screenshot shows the Oracle SQL Developer interface with a query window titled 'PSUG-MW'. The query is as follows:

```
SELECT DISTINCT
  grade_level,
  home_room
FROM
  students
ORDER BY
  grade_level,
  home_room
```

The query results are displayed in a table with 7 rows. The first two columns are 'GRADE_LEVEL' and 'HOME_ROOM'. The results are:

	GRADE_LEVEL	HOME_ROOM
1	0	Abbot
2	0	Adair
3	0	Allen
4	0	Arnold
5	0	(null)
6	1	Barker
7	1	Blain

Tables



Tables

- Everything is stored in tables
 - Students
 - Schools
 - Courses
 - Sections
 - Terms
- Same tables and columns as DDE

Schools Table

The screenshot shows the Oracle SQL Developer interface. The main window is titled "Oracle SQL Developer : PSUG-MW". The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations, execution, and formatting. The left sidebar shows the "Connections" and "Reports" panels. The main workspace is divided into two tabs: "Worksheet" and "Query Builder". The "Worksheet" tab is active, displaying a SQL query:

```
SELECT
  name
FROM
  schools
```

The "Query Result" panel at the bottom shows the results of the query. It indicates that all rows were fetched in 0.033 seconds. The results are displayed in a table with a single column named "NAME".

NAME
1 Graduated Students
2 Apple Grove High School 3
3 Apple Grove High School
4 Apple Grove High School 2
5 Scheduling High School 1
6 Cherry Hill Middle School
7 Washington Elementary
8 ...

The status bar at the bottom indicates the current cursor position: "Line 4 Column 10 | Insert | Modified | Windows: C".

Courses Table

The screenshot shows the Oracle SQL Developer interface with a connection to 'PSUG-MW'. The 'Worksheet' tab is active, displaying the following SQL query:

```
SELECT
  course_name,
  course_number,
  credittype
FROM
  courses
```

The 'Query Result' tab is also active, showing the results of the query. The status bar indicates 'Fetched 50 rows in 0.057 seconds'. The results are displayed in a table with three columns: COURSE_NAME, COURSE_NUMBER, and CREDITTYPE.

	COURSE_NAME	COURSE_NUMBER	CREDITTYPE
1	Speech	12005B	ENG
2	Journalism	12006A	ENG
3	Journalism	12006B	ENG
4	Creative Writing	12007A	ENG
5	Creative Writing	12007B	ENG
6	English Lit & Comp	12008A	ENG
7	English Lit & Comp	12008B	ENG
8	English Lit & Comp	12008C	ENG

The status bar at the bottom indicates 'Line 4 Column 13 | Insert | Modified | Windows: C'.

Courses Table

The screenshot shows the Oracle SQL Developer interface with a query executed in the 'PSUG-MW' workspace. The query selects course details from the 'courses' table, filtered by 'MAT' credit type. The results pane shows 28 rows of data.

Query:

```
SELECT
  course_name,
  course_number,
  credittype
FROM
  courses
WHERE
  credittype = 'MAT'
```

Query Result: All Rows Fetched: 28 in 0.035 seconds

	COURSE_NAME	COURSE_NUMBER	CREDITTYPE
1	Algebra	MAT1100	MAT
2	AP Calculus	MAT3000	MAT
3	Calculus	MAT2100	MAT
4	Consumer Math	MAT1000	MAT
5	Consumer Math	10001A	MAT
6	Consumer Math	10001B	MAT
7	Pre-Algebra	10002A	MAT
8

Line 8 Column 21 | Insert | Modified | Windows: CF

Tables and Columns to Know

- **Students**

- grade_level
- enroll_status
 - Enumerated ([KB 5967](#))
 - -1 = Pre-registered
 - 0 = Active
 - 2 = Transferred
 - 3 = Graduated
- entrydate
- exitdate
- home_room
- schoolid (linked to the schools table)

Tables and Columns to Know

- **Schools**

- school_number
- name

- **Terms**

- schoolid
- firstday
- lastday
- yearid

Tables and Columns to Know

- **Courses**
 - course_number
 - course_name
- **CC** (Current Courses)
 - studentid
 - course_number
 - sectionid
 - termid
 - dateenrolled
 - dateleft
 - expression

Tables and Columns to Know

- **StoredGrades**
 - studentid
 - course_number
 - grade
 - gradescale_name
 - sectionid
 - storecode
 - termid
 - gpa_points
 - gpa_addedvalue
 - potentialcrhrs
 - earnedcrhrs

Tables and Columns to Know

- **Sections**

- course_number
- expression
- schoolid
- section_number
- teacher
- termid

Tables and Columns

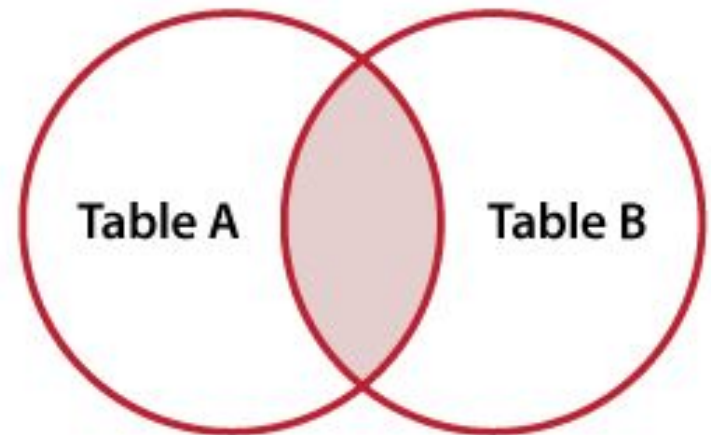
- Data Dictionary



- Any other data you want to extract?

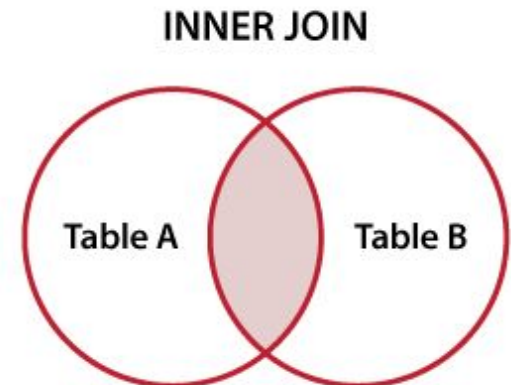
JOINS

INNER JOIN



JOINS

- INNER JOINS are the default type and are the most intuitive
 - Matches rows from one table with rows in another table
 - Returns fields from one or both tables in the result set
 - Can be strung together to combine multiple tables into one result table



JOIN Concept

lastfirst	schoolid
Ackerman, Stan	600
Adair, Brandon	100
Adams, Brandon G	700
Adams, Corby	300
Adams, Corby	200
...	...

Students

JOIN syntax

school_number	name
100	Apple Grove High School
200	Apple Grove High School 2
300	Apple Grove High School 3
600	Cherry Hill Middle School
700	Washington Elementary
...	...

Schools

lastfirst	name
Ackerman, Stan	Cherry Hill Middle School
Adair, Brandon	Apple Grove High School
Adams, Brandon G	Washington Elementary
Adams, Corby	Apple Grove High School 3
Adams, Corby	Apple Grove High School 2
...	...

JOINS

- INNER is default type of JOIN
 - Only returns records matched in BOTH tables
- Can create ambiguous column names
 - id appears in almost every table
 - Recommended to use table.column notation to reference columns when using any JOIN

INNER JOINS

- Basic syntax:

- `SELECT`
 `fields`
`FROM`
 `table1`
 `JOIN table2`
 `ON table1.table2id = table2.id`
- Without the `ON` condition, every row from `table1` will be multiplied by every row from `table 2`.
 - This is probably not what you want.

INNER JOINS

- For example, imagine that you want a list of student names and the school they attend
- Student name is found in the students table
- School name is in the schools table
- How do we match?
 - `students.schoolid` is linked to `schools.school_number`
 - We will join students to schools using this unique identifier and return only the columns we want

JOIN Concept

lastfirst	schoolid
Ackerman, Stan	600
Adair, Brandon	100
Adams, Brandon G	700
Adams, Corby	300
Adams, Corby	200
...	...

Students

JOIN syntax

school_number	name
100	Apple Grove High School
200	Apple Grove High School 2
300	Apple Grove High School 3
600	Cherry Hill Middle School
700	Washington Elementary
...	...

Schools

lastfirst	name
Ackerman, Stan	Cherry Hill Middle School
Adair, Brandon	Apple Grove High School
Adams, Brandon G	Washington Elementary
Adams, Corby	Apple Grove High School 3
Adams, Corby	Apple Grove High School 2
...	...

lastfirst	schoolid
Ackerman, Stan	600
Adair, Brandon	100
Adams, Brandon G	700
Adams, Corby	300
Adams, Corby	200
...	...

Students

```

SELECT
students.lastfirst,
schools.name

FROM
students
JOIN schools
ON students.schoolid
= schools.school_number

```



lastfirst	name
Ackerman, Stan	Cherry Hill Middle School
Adair, Brandon	Apple Grove High School
Adams, Brandon G	Washington Elementary
Adams, Corby	Apple Grove High School 3
Adams, Corby	Apple Grove High School 2
...	...

school_number	name
100	Apple Grove High School
200	Apple Grove High School 2
300	Apple Grove High School 3
600	Cherry Hill Middle School
700	Washington Elementary
...	...

Schools

INNER JOIN

The screenshot shows the Oracle SQL Developer interface. The main window displays an SQL query in the Worksheet tab. The query is an inner join between the 'students' and 'schools' tables, filtered by 'students.schoolid' values of 100, 200, 300, 600, and 700, and ordered by 'lastfirst'.

```
SELECT
  students.lastfirst,
  schools.name
FROM
  students
JOIN schools ON students.schoolid = schools.school_number
WHERE
  students.schoolid in (100,200,300,600,700)
ORDER BY
  lastfirst
```

Below the query editor, the 'Query Result' tab is active, showing the results of the query. It indicates that 50 rows were fetched in 0.044 seconds. The results are displayed in a table with two columns: 'LASTFIRST' and 'NAME'.

	LASTFIRST	NAME
1	Ackerman, Stan	Cherry Hill Middle School
2	Adair, Brandon	Apple Grove High School
3	Adams, Brandon G	Washington Elementary
4	Adams, Corby	Apple Grove High School 3
5	Adams, Corby	Apple Grove High School 2
6	Adams, Corby	Apple Grove High School

The status bar at the bottom indicates 'Line 6 Column 60 | Insert | Modified | Windows: CF'.

Current schedule example

```
SELECT
    courses.course_name,
    teachers.lastfirst,
    cc.expression,
    students.lastfirst,
    cc.dateenrolled,
    cc.dateleft
FROM
    cc
    JOIN students ON cc.studentid = students.id
    JOIN courses ON UPPER(cc.course_number) =
                    UPPER (courses.course_number)
    JOIN sections on cc.sectionid = sections.id
    JOIN teachers on sections.teacher = teachers.id
WHERE
    cc.dateenrolled <= sysdate
    AND cc.dateleft > sysdate
```

Current schedule example

The screenshot shows the Oracle SQL Developer interface. The title bar indicates the file path: C:\Users\aplarsen\AppData\Roaming\SQL Developer\Untitled3.sql. The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains icons for Start Page, PSUG-MW, and several untitled SQL files. The main workspace is divided into two tabs: Worksheet and Query Builder. The Worksheet tab is active, displaying a SQL query. The Query Result tab is also visible, showing the results of the query. The status bar at the bottom indicates the current position: Line 6 Column 21, and provides options for Insert, Modified, and Windows.

```
FROM
cc
JOIN students ON cc.studentid = students.id
JOIN courses ON UPPER(cc.course_number) = UPPER (courses.course_number)
JOIN sections on cc.sectionid = sections.id
JOIN teachers on sections.teacher = teachers.id
WHERE
cc.dateenrolled <= sysdate
AND cc.dateleft > sysdate
```

Query Result x

SQL | Fetched 50 rows in 1.088 seconds

	COURSE_NAME	LASTFIRST	EXPRESSION	LASTFIRST_1	DATEENROLLED	DATELEFT
1	English 2	Mantezimbe, Yoruba	3 (B)	Auclair, Hattie	27-MAY-13	23-MAY-14
2	Weight Training	Rutter, Stephen D	1 (B)	Auclair, Hattie	30-NOV-13	23-MAY-14
3	Sociology	Luhman, Jason Q	3 (A)	Auclair, Hattie	27-MAY-13	23-MAY-14
4	Intro to Art	Russell, Chris C	2 (A)	Auclair, Hattie	30-NOV-13	23-MAY-14
5	Health 11	Myers, Winston X	2 (B)	Auclair, Hattie	30-NOV-13	23-MAY-14
6	Chemistry 1	Dombeck, Bartolomeu E	4 (A-B)	Auclair, Hattie	30-NOV-13	23-MAY-14

Line 6 Column 21 | Insert | Modified | Windows: CF

Current grades example

```
SELECT
-- -- --
    students.last_name || ', ' || students.first_name student,
    pgfinalgrades.finalgradename,
    grade,
    percent,
    courses.course_number,
    courses.course_name,
    teachers.last_name teacher
FROM
    pgfinalgrades
    JOIN sections on pgfinalgrades.sectionid = sections.id
    JOIN teachers on sections.teacher = teachers.id
    JOIN students on pgfinalgrades.studentid = students.id
    JOIN courses on sections.course_number = courses.course_number
WHERE
    pgfinalgrades.finalgradename = 'S2'
    AND sections.termid >= 2800
ORDER BY
    students.grade_level,
    students.lastfirst,
    courses.course_name
```

Current grades example

The screenshot shows the Oracle SQL Developer interface. The title bar indicates the connection is 'ps9c.clgpsug.com'. The menu bar includes File, Edit, View, Navigate, Run, Source, Team, Tools, Window, and Help. The toolbar contains various icons for file operations and development. The 'Connections' pane on the left shows the current connection. The main editor displays a SQL query in the 'Query Builder' tab:

```
pgfinalgrades.finalgradename = 'S2'  
AND sections.termid >= 2500  
ORDER BY  
students.grade_level,  
students.lastfirst,  
courses.course_name
```

Below the editor, the 'Query Result' tab shows the results of the query. It indicates that 50 rows were fetched in 0.737 seconds. The results are displayed in a table with the following columns: STUDENT, FINALGRADENAME, GRADE, PERCENT, COURSE_NUMBER, COURSE_NAME, and TEACHER.

STUDENT	FINALGRADENAME	GRADE	PERCENT	COURSE_NUMBER	COURSE_NAME	TEACHER
1 Alu, Matthew	S2	B	86	EL2000	Art	Gonzales
2 Alu, Matthew	S2	B	81	EL1000	General Music	Kim
3 Alu, Matthew	S2	--	0	HR	Homeroom	Abbot
4 Alu, Matthew	S2	A	100	EL1	Kindergarten	Abbot
5 Alu, Matthew	S2	F	9	EL3000	Physical Education	Riley
6 Andersen, Anthony	S2	D	69	EL2000	Art	Gonzales
7 Andersen, Anthony	S2	A	95	EL1000	General Music	Kim

The status bar at the bottom shows the current cursor position as 'Line 17 Column 34' and includes buttons for 'Insert', 'Modified', and 'Windows: C'.

Table aliasing



- Some people like to alias their tables:

```
SELECT
```

```
    st.grade_level grade,
```

```
    sch.name school,
```

```
    st.lastfirst student
```

```
FROM
```

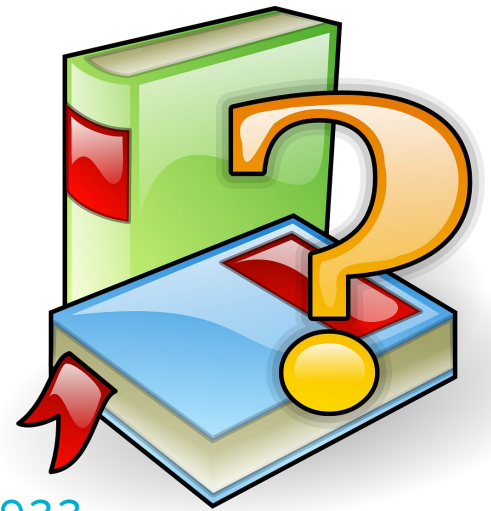
```
    students st
```

```
    JOIN schools sch ON st.schoolid = sch.school_number
```

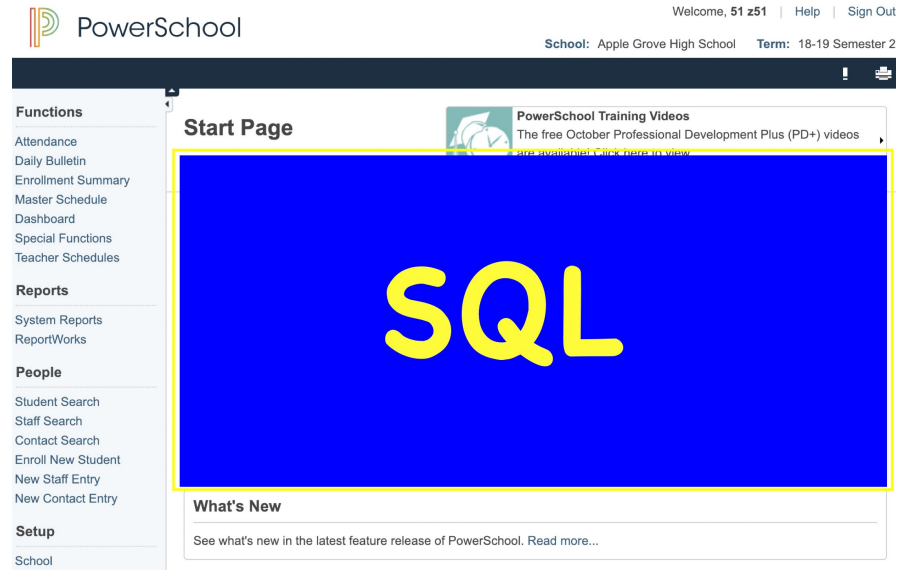
- This can make your SQL more difficult to read and troubleshoot.
- It also makes it harder to reuse code in the future.
- Full table names are more typing, but you never have to decode your work later

Resources

- PowerSource Data Dictionaries
 - <https://support.powerschool.com/dir/5933>
- PowerSource Oracle Account Passwords
 - <https://support.powerschool.com/article/55006>
- *Learning SQL* from O'Reilly:
 - <http://shop.oreilly.com/product/9780596520847.do>
- *SQL Pocket Guide* from O'Reilly:
 - <http://shop.oreilly.com/product/0636920013471.do>
- Tech on the Net
 - <http://www.techonthenet.com/sql/index.php>
- W3 Schools
 - <http://www.w3schools.com/sql>



How Can I Use SQL inside PowerSchool?



How can I use SQL inside PS?

- **Tlist_sql in customizations**

```
~[tlist_sql;
```

```
{query}
```

```
;alternatecolor;nonemessage={none_message}]
```

```
{row_template}
```

```
[/tlist_sql]
```

Tlist_sql

```
~[tlist_sql;  
SELECT lastfirst, grade_level  
FROM students  
WHERE  
enroll_status = 0 AND  
schoolid = ~(curschoolid)  
ORDER BY lastfirst  
;alternatecolor;nonemessage=No Records Found]  
<tr>  
<td>  
~(lastfirst;t)  
</td>  
<td>  
~(grade_level;l)  
</td>  
</tr>  
[/tlist_sql]
```



Customize Page

02/10/2019, 10:14:01PM (Active)

Extract to Keys

Preview Keys

Preview

Save Draft

Publish

02/10/119, 10:17:13PM (Draft)

Snippets

Templates

studentlist.html

Editable

```
20 <!-- start of content area -->
21 <div class="box-round">
22
23 <!-- This is a standards driven table. there are no styles, borders, widths and ther
24 <table border="0" cellpadding="0" cellspacing="0" class="grid" id="tableUniqueID">
25 <thead>
26 <tr>
27 <th>Student</th>
28 <th>Grade</th>
29 </tr>
30 </thead>
31 <tbody>
32 ~[tlist_sql;
33 SELECT lastfirst, grade_level
34 FROM students
35 WHERE
36 enroll_status = 0 AND
37 schoolid = ~(curschoolid)
38 ORDER BY
39 lastfirst
40 ;alternatecolor;nonemessage=No Records Found]
41 <tr>
42 <td>
43 ~(lastfirst;t)
44 </td>
45 <td>
46
```

Your breadcrumbs go here

**Functions**[Attendance](#)
[Daily Bulletin](#)
[Enrollment Summary](#)
[Master Schedule](#)
[Dashboard](#)
[Special Functions](#)
[Teacher Schedules](#)**Reports**[System Reports](#)
[ReportWorks](#)**People**[Student Search](#)
[Staff Search](#)
[Contact Search](#)

Student Roster

Student	Grade
Baker, Cole	11
Barlow, Elysa F	10
Barlow, Gayle D	12
Bartlett, Jared	11
Baumann, Amber B	12
Bennett, Cody B	9
Bevan, Adam C	10
Birkeland, Matthew S	11
Braun, Jessica	11

```

<form action="gradelevelroster2.html"
method="GET">
<table class="linkDescList">
  <tr>
    <th>Option</th>
    <th>Value</th>
  </tr>
  <tr>
    <td>Grade Level</td>
    <td>
      <select name="grade_level">
        <option value=""><option
value="-2">PK3<option value="-1">PK4<option
value="0">K<option>1<option>2<option>3<option>
4<option>5<option>6<option>7<option>8<option>9
<option>10<option>11<option>12
      </select>
    </td>
  </tr>
</table>
<div class="button-row">~[submitbutton]</div>
</form>

```

Grade Level Roster

Choose a Grade Level

Option	Value
Grade Level	<div> <div>✓</div> <div>PK3</div> <div>PK4</div> <div>K</div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> <div>8</div> <div>9</div> <div>10</div> <div>11</div> <div>12</div> </div>

Submit

Example of sending a gpv to a custom tlist_sql report

https://.../gradelevelroster2.html?grade_level=3&btnSubmit=

```
<table>
<tr>
  <th>Grade</th><th>Last</th><th>First</th>
</tr>
~[tlist_sql;
  select grade_level, last_name, first_name
  from students
  where
    ~(curschoolid) in (0, schoolid)
    and enroll_status = 0
    and grade_level = ~(gpv.grade_level)
  order by lastfirst
;]
<tr>
  <td>~(grade_level)</td>
  <td>~(last_name)</td>
  <td>~(first_name)</td>
</tr>
[/tlist_sql]
</table>
```

Grade	Last	First
3	Adams	Jennifer
3	Ainsworth	Cole
3	Berg	Stephanie
3	Birkeland	Casey
3	Bourassa	Vanessa
3	Bradshaw	Charidee
3	Brakke	Rosally
3	Brandt	Alan
3	Braton	Kelly
3	Brooks	Nicholas

Example of receiving a gpv in a custom report (grade_level=3)

Tlist_SQL

- Tlist_sql modifiers for the “variables”
 - ;l = “Long Integer” (numbers)
 - ;t = “Text”
 - ;d = “Date”
- <https://support.powerschool.com/developer/#/page/ps-html-tags>

Current Selection

When working with the Students table, you can query the current selection of students by adding the following WHERE clause:

```
WHERE dcid IN (SELECT dcid FROM  
~[temp.table.current.selection:students])
```

Deprecated?

support.powerschool.com/developer

TLIST_SQL

"TLIST_SQL" is the name of a tag that executes SQL right in the page and returns the results. It is **STRONGLY DEPRECATED** in new development work for several reasons: if not used carefully it can easily introduce security holes, reflects improper layering, and because there is almost always a better way to do the same operation. (See [PowerQuery reference](#).)

What does it mean when something is deprecated? ^

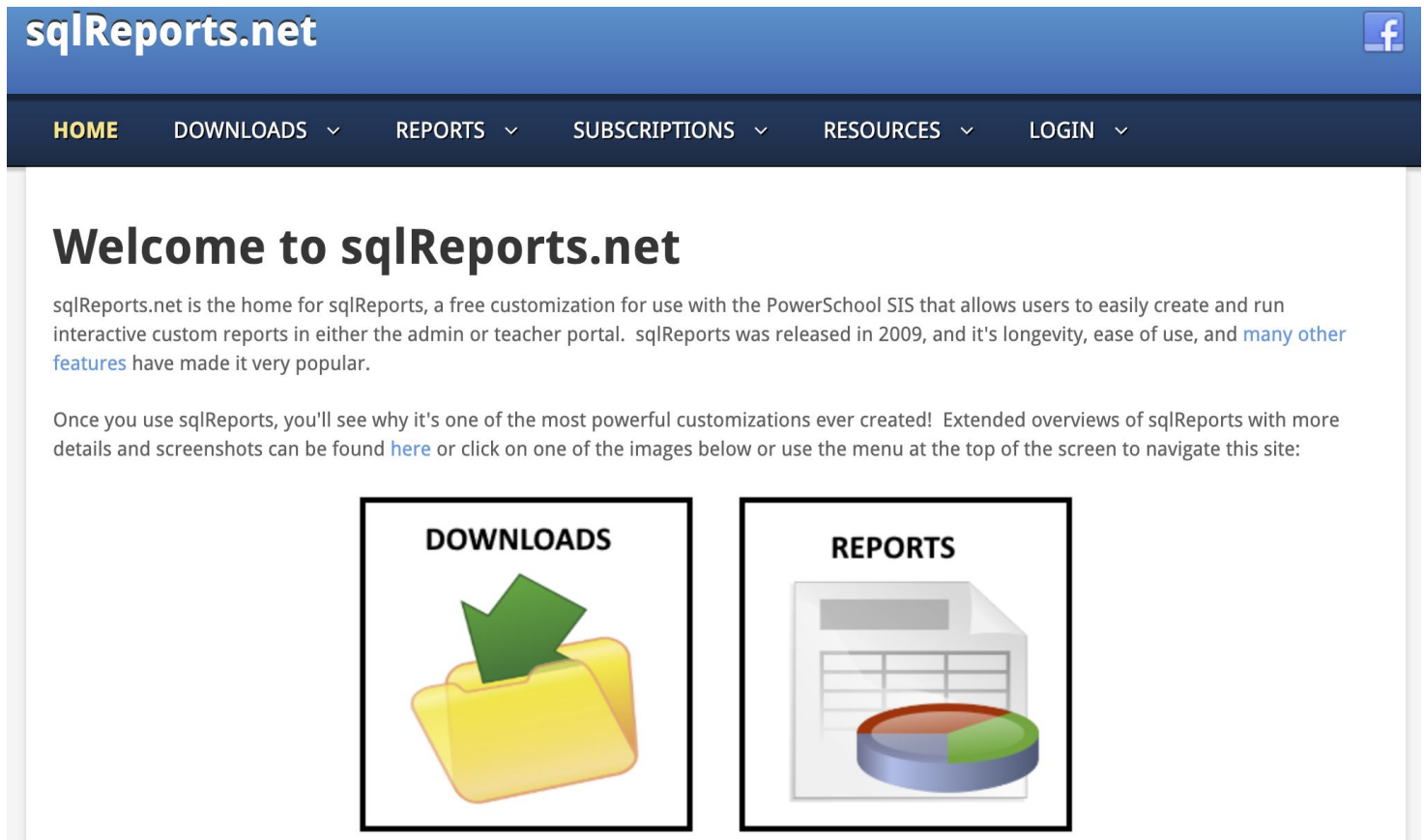
In the world of software development, "**deprecated**" refers to functions or elements that are in the process of being replaced by newer ones. The term comes from the word "deprecate," which **means** to disapprove of **something**. Feb 25, 2011

How can I use SQL inside PS?

- Enterprise Reporting (APEX)
 - <https://support.powerschool.com/article/77492>
 - [psugcal.org/index.php?title=Enterprise Reporting](https://psugcal.org/index.php?title=Enterprise_Reporting)

How can I use SQL inside PS?

- sqlReports



History of sqlReports

- Dean Dahlvang
 - Released March 2009
 - Through version 3
 - "an ad hoc reporting add-on that can turn simple to medium sql queries in to web based reports in PowerSchool"
- Matt Freund
 - In 2013, Dean turned over development to Matt Freund
 - Coding assistance from Bob McGregor
 - sqlReports 4 was born and first released June, 2013





What it does

- Create new reports in PS that use SQL, hence the name sqlReports.
- Reports act like custom pages, but sqlReports keeps everything in one central location and makes a menu for you.
- Admin or Teacher portal.
- Control access based on security groups.
- The menu system makes it easy to create new reports, edit reports, and is an easy way for your users to find reports.

What it does

[Create a new sqlReport](#) - [Import a new sqlReport](#) - [sqlReports Help](#)

[Expand All](#) [Collapse All](#)

Report	Title	
▸ Attendance		
▸ labels		
▸ sqlExports		
▸ sqlReportsQ		
Bus Information	Bus Information	
Class Enrollments - search by date	Class Enrollments - search by date	
CPM - Custom Files Information	CPM - Custom Files Information	
Current Grades Report - No Minimum	Current Grades Report - No Minimum	

What it does

- Reports can be simple tabular reports with buttons to copy, export, or print the results
- Reports can be single series or multi-series charts, with over 25 charts to choose from.

The SQL Behind the Report

--- Example of the SQL area of a report setup, from a free report called "Current Grades - No Minimum"

SQL Query

Follow regular SQL syntax

Can include parameters, which can be either system variables like ~(curschoolid) or ones that you create, such as the param1 reference in the Where.

```
SELECT s.dcid, s.lastfirst, s.grade_level, pgf.Grade, pgf.Percent, c.Course_Name,
u.LastFirst, to_char(pgf.LastGradeUpdate,'MM/DD/YYYY')

    from Students s
~[if#cursel.%param2%=Yes]
        INNER JOIN ~[temp.table.current.selection:students] stusel ON
stusel.dcid=s.dcid
[/if#cursel]
        INNER JOIN PGFinalGrades pgf ON s.ID = pgf.StudentID
        INNER JOIN Sections sec ON pgf.SectionID = sec.ID
        INNER JOIN Courses c ON upper(sec.Course_Number) =
upper(c.Course_Number)
        INNER JOIN SchoolStaff ss on ss.id = sec.Teacher
        INNER JOIN Users u on u.dcid = ss.users_dcid
        INNER JOIN CC cc on cc.SectionID = sec.ID AND pgf.StudentID =
cc.StudentID

    WHERE s.enroll_status = 0
        and s.schoolid = ~(curschoolid)
        and pgf.FinalGradeName = '%param1%'
        and cc.Termid like '~(curyearid)%%'
        and pgf.Grade IN (%param3%)

    order by s.lastfirst
```

Running the Report

- Users click on the report and run it. If you have any parameters, they will be prompted to enter them when they run the report. In the example below, the user is prompted for three things - Term, whether or not to Use Current Selection, and a place to Enter Grades. Same report, multiple uses!

Label	Value
Name	Current Grades Report - No Minimum
Description	Can be run for all active students at this school or just the current selection. For the grade parameter, enter each grade in single quotes, and if there's more than one grade, create a comma-delimited list. For example, either 'F' by itself or 'D','D+','D-','F' for D's and F.
Directions	
Term	<input type="text" value="Q1"/>
Use Current Selection?	<input type="text" value="No"/>
Enter Grades	<input type="text" value="'F'"/>

Submit

Results Page

Current Grades Report - No Minimum

Parameters - Term: Q1 Use Current Selection?: No Enter Grades: 'F'

Make Current Selection

Copy

CSV

TAB

Print

PDF

Student

Grade Level

Grade

Course

Teacher

Search:

Student	Grade Level	Grade	Percent	Course	Teacher	Last Update
Aikinson, Andy G	12	F	42	Calculus	Bryant, Renata L	01/11/2017
Allen, Victor C	11	F	57	Physics	Zuker, Donald K	06/25/2015
Allred, Alfred P	10	F	50	Phys Ed 10	Maxedon, Johnathon	06/18/2015
Allred, Alfred P	10	F	40	Concert Choir	Berndt, Gordie E	06/25/2015

- Option to copy, export, or print the data
- Make students the current selection.
- The results can be narrowed down by a search or by column filtering.
- Action buttons honor filtering.

Advantages

- No Need to Know How to Create Custom Pages
- No Need to Know How to Code Charts
- It's Available for Admins or Teachers
- Saves Time
- Continuous Development
- Popularity

Continue at sqlReports.net

- Check out additional add-ons
 - sqlReports - Teacher Portal
 - sqlCharts
 - sqlExports
 - sqlReportsQ
 - Saved Params
 - Custom Links
 - sqlFormLetters
- Reports section with free and premium reports you can import
- Article, Documentation and Blog Posts