

Oracle Academy

Database Programming with SQL

Instructor Resource Guide

Notes for PowerPoint Slides:

SECTION 4 LESSON 1 – Putting It All Together

Slide 1: Putting It All Together

Lesson Preparation

Use Internet resources to locate information about job predictions for the future (keyword search: "future careers" and "predicted future jobs").

The U.S. Department of Labor website provides excellent information about all aspects of the labor market and "tomorrow's jobs" in the United States.

Additional sites include:

<http://www.bls.gov/news.release/nlsoy.t01.htm>

<ftp://ftp.bls.gov/pub/news.release/nlsoy.txt>

United Kingdom: Department for Education and Skills, www.london.edu

Careers in Europe:

www.hobsons.com, <http://www.exp.ie/advice/eurograduate.html>,

<http://www.iht.com/IHT/SUP/052799/car01.html>

<http://doctorjob.com/about/>

Additional sites include:

http://www.careerplanner.com/Career-Articles/Top_Jobs.htm

http://www.readersdigest.ca/mag/2000/06/living_job.html

Slide 2: What Will I Learn?

No instructor notes for this slide

Slide 3: Why Learn It?

No instructor notes for this slide

Slide 4: Tell Me / Show Me – Think of jobs that are now extinct or ...

What to Watch For

Explain that there is no crystal ball to predict what the future holds. Life-changing events such as the World Trade Center disaster, wars, stock-market speculation, environmental disasters such as earthquakes, volcanoes, hurricanes, etc. can have a radical impact on world economies. That aside, planning is still the best way to avoid becoming obsolete in the job market.

Slide 5: Tell Me / Show Me – In 1990 the average number of career changes ...

No instructor notes for this slide

Slide 6: Tell Me / Show Me – Global markets: businesses have ...

No instructor notes for this slide

Slide 7: Summary – In this lesson you have learned to:
No instructor notes for this slide

Slide 8: Summary - Practice Guide
No instructor notes for this slide

SECTION 4 LESSON 2 – Cross Joins and Natural Joins

Slide 1: Cross Joins and Natural Joins

Lesson Preparation

The joins in this and the next two lessons will use the ANSI/ISO SQL: 1999 syntax. Write the following list on the board to give students an overview of what will be covered in these lessons. The ANSI/ISO SQL: 1999 join terms are: cross join, natural join, USING clause, outer join.

Slide 2: What Will I Learn?

No instructor notes for this slide

Slide 3: Why Learn It?

Students will confuse which joins are Oracle proprietary joins and which are ANSI/ISO SQL: 1999 SQL. The important point is for students to be able to choose correct joins and apply them correctly.

Slide 4: Tell Me / Show Me – ANSI

Discuss the importance of standards. Ask students what other standards they count on in everyday life? Possible answers include: time standard, weights and measures, international signs, Internet addresses.

Slide 5: Tell Me / Show Me – SQL

Discuss with students: "Why is it important for Oracle to comply with the current ANSI standard for SQL? Why is it important to comply with any standard? For example, a measuring cup in one store is the same size as a measuring cup in another, or the 185/60HR14 tire size in one store is the same tire size in another store.

Slide 6: Tell Me / Show Me – NATURAL JOIN

No instructor notes for this slide

Slide 7: Tell Me / Show Me – NATURAL JOIN (continued)

As shown in the sample code, when using ...

Ask students: which column or columns have the two tables been joined on ? Answer: song_id, because it is the only column with the same name and data type in both tables.

Slide 8: Tell Me / Show Me – NATURAL JOIN (continued)

Here is another example:

The tables have been joined on client_number.

Slide 9: Tell Me / Show Me – CROSS JOIN

Recall the Cartesian Product...

Relate a cross join to other examples:

How many ways are there to arrange the following numbers: 1 2 3?

Answer: 6

How many different combinations of outfits are possible with a red sweater and a blue t-shirt with blue jeans, hiking shorts, and a green hat?

Answers will vary.

Slide 10: Tell Me / Show Me – CROSS JOIN

Cross Join Example:

In the example cross join, ask students to count the number of rows in both tables. The number of rows returned by the query will be the product of the number of rows in each table: $2 * 6 = 12$.

Slide 11: Tell Me /Show Me - Terminology

CROSS JOIN>Returns the Cartesian product from two tables.

NATURAL JOIN-Joins two tables based on the same column name.

Slide 12: Summary – In this lesson you have learned to:

No instructor notes for this slide

Slide 13: Summary - Practice Guide

No instructor notes for this slide

SECTION 4 LESSON 3 – Join Clauses

Slide 1: Join Clauses

What to Watch For

Joins and the different applications are challenging for students. Add practice whenever possible. Help students organize the Oracle proprietary joins and the ANSI SQL joins and the similarities and differences between them.

Connections

Relate joins to other examples:

For security reasons, usernames and passwords are stored in different tables. For a user to have computer access, the username and password must be checked. A join is a possible way to accomplish this task.

Internet online shopping uses the "shopping cart" feature. The software is able to bring together information, such as customer name, payment method, and items ordered, from several tables to display a completed order on the screen.

Slide 2: What Will I Learn?

The following websites have good tutorials on joins that can help students read other explanations for all the complicated join statements.

<http://www.databasejournal.com/>

<http://www.1keydata.com/sql/sql.html>

<http://otn.oracle.com/oramag/oracle/01-nov/o61sql.html>

Slide 3: Why Learn It?

Discuss the graphic. To join these two tables, there needs to be a column shared by each table.

D_THEMES code column appears in D_EVENTS theme_code column.

Slide 4: Tell Me / Show Me – USING CLAUSE

In a natural join, if the tables ...

A USING clause is often preferred to a natural join even when the columns have the same data type as well as the same name, because it states clearly exactly which join column is being used.

Slide 5: Tell Me / Show Me – USING CLAUSE (continued)

The query shown is an example...

Ask students to rewrite the JOIN USING example as an equijoin. Note that qualifiers can be used.

```
SELECT c.client_number, c.first_name, c.last_name, e.event_date
FROM d_clients c, d_events e
WHERE c.client_number = e.client_number;
```

Slide 6: Tell Me / Show Me – USING CLAUSE (continued)

The USING clause allows us to use ...

No instructor notes for this slide

Slide 7: Tell Me / Show Me – ON CLAUSE

What if the columns to be joined have ...

No instructor notes for this slide

Slide 8: Tell Me / Show Me – ON CLAUSE (continued)

In this example, the ON clause is ..

No instructor notes for this slide

Slide 9: Tell Me / Show Me – ON CLAUSE (continued)

Here is the same query with a WHERE ...

No instructor notes for this slide

Slide 10: Tell Me / Show Me –JOINING THREE TABLES

No instructor notes for this slide

Slide 11: Tell Me / Show Me –JOINING THREE TABLES (continued)

Point out that the order in which the tables are joined is important. Could we use:

```
SELECT last_name, event_date, t.description
```

```
FROM d_clients c JOIN d_themes t
```

```
ON (.....)
```

```
JOIN d_events e
```

```
USING (.....); ?
```

Answer: No, because there is no relationship between d_clients and d_themes. What columns would we join them on ?

Slide 12: Tell Me / Show Me – Comparing Oracle Proprietary

No instructor notes for this slide

Slide 13: Tell Me /Show Me - Terminology

ON clause-Allows a natural join based on an arbitrary condition or to specify columns to join.

USING clause-Performs an equijoin based on the same specified column name

Slide 14: Summary – In this lesson you have learned to:

No instructor notes for this slide

Slide 15: Summary - Practice Guide

No instructor notes for this slide

SECTION4 LESSON 4 – Inner versus Outer Joins

Slide 1: Inner versus Outer Joins

Lesson Preparation

Review the Oracle proprietary outer join and compare it to the ANSI SQL left and right outer joins. Compare the location of the (+) sign versus the word LEFT or RIGHT in the ANSI SQL joins. Draw a seesaw on the board, and add a (+) sign to the side of the board that is up in the air. It's missing something (weight), so we need to add a (+) to it.

Slide 2: What Will I Learn?

No instructor notes for this slide

Slide 3: Why Learn It?

No instructor notes for this slide

Slide 4: Tell Me / Show Me – INNER AND OUTER JOINS

No instructor notes for this slide

Slide 5: Tell Me / Show Me – LEFT AND RIGHT OUTER JOINS

In the example shown of a left outer join...

Point out that Grant is not assigned to a department.

Note that we could have used USING instead of ON in this case: USING
(department_id)

The column headings DEPARTMENT_ID and DEPARTMENT_NAME have been abbreviated in this and the following slides, to save space on the slide.

Slide 6: Tell Me / Show Me – LEFT AND RIGHT OUTER JOINS (continued)

This right outer join would return...

Point out that we could get the same results by changing RIGHT to LEFT and reversing the order of table names:

.... FROM departments d LEFT OUTER JOIN employees e

Slide 7: Tell Me / Show Me – FULL OUTER JOIN

It is possible to create a join condition to retrieve...

No instructor note for this slide

Slide 8: Tell Me / Show Me – FULL OUTER JOIN (continued)

The example shown is a full outer join.

Ask students: if this query had instead been created to use an inner join (so that Grant and Contracting would not appear in the output), how could the result of the query be misleading? Possible answers include: whoever reads the result could think that employee Grant and the Contracting department did not exist.

Slide 9: Tell Me / Show Me – Construct a join to display a list of Global Fast...

Ask students how to modify the join statement to include only orders from December 1, 2002, to December 31, 2004, inclusive. Answer: WHERE o.order_date BETWEEN '01-JAN-02' AND '31-DEC-04';

Discuss with students, "Why is it important, from a business perspective, to be able to join data from multiple sources in so many different ways?" In the real world, you often count on outer joins for checking your data. A business rule may state, "You must enter all new employees and then assign them to a department." What if the system allowed employees to be entered without departments, even though the business rule requires them to have departments? Well, first, the system would not be perfect. Someone didn't design it properly with the business needs in mind. However, as the programmer, you may be asked to find those employees without departments and produce a report so that a manager can assign them to a department and correct the data!

Slide 10: Tell Me /Show Me - Terminology

FULL OUTER JOIN-Performs a join on two tables, retrieves all the rows in the Left table, even if there is no match in the Right table. It also retrieves all the rows in the Right table, even if there is no match in the Left table.

Outer join-A join that returns the unmatched rows as well as matched rows

LEFT OUTER JOIN-Performs a join on two tables, retrieves all the rows in the Left table even if there is no match in the Right table.

RIGHT OUTER JOIN-performs a join on two tables, retrieves all the rows in the Right table even if there is no match in the Left table.

Inner join-A join of two or more tables that return only matched rows

Slide 11: Summary – In this lesson you have learned to:

No instructor notes for this slide

Slide 12: Summary - Practice Guide

No instructor notes for this slide

Notes For Practice Activities:

Putting It All Together S04 L01

Try It / Solve It

Individual Responses

Cross Joins and Natural Joins S04 L02

Vocabulary

CROSS JOIN Returns the Cartesian product from two tables.

NATURAL JOIN Joins two tables based on the same column name.

Try It / Solve It

1. SELECT last_name, department_name
FROM employees
CROSS JOIN departments;

2. A Cartesian product.

3. SELECT department_id, department_name, location_id, city
FROM departments
NATURAL JOIN locations;

4. SELECT d.department_id, d.department_name, d.location_id, l.city
FROM departments d, locations l
WHERE d.location_id = l.location_id;

5. SELECT department_id, department_name, location_id, city
FROM departments
NATURAL JOIN locations
WHERE department_id IN (20,50);

6. SELECT type_code, description, title FROM d_songs, d_types
WHERE d_songs.type_code = d_types.code
AND type_code BETWEEN 70 and 80;

7. WHERE

8. FROM

9. ANSI syntax may be used with any database

10. alias or table name

11. FROM

Join Clauses S04 L03

Vocabulary

<u>ON clause</u>	Allows a natural join based on an arbitrary condition or to specify columns to join.
<u>USING clause</u>	Performs an equijoin based on the same specified column name

Try It / Solve It

1. SELECT l.city, d.department_name
FROM locations l JOIN departments d USING (location_id)
WHERE location_id = 1400;
2. SELECT song_id, cd_number, title, comments
FROM d_play_list_items JOIN d_track_listings USING (song_id) JOIN d_cds USING
(cd_number);
3. SELECT l.city, d.department_name, location_id, d.department_id
FROM locations l JOIN departments d USING (location_id)
WHERE city = 'Seattle'
AND department_id IN (10, 20, 30);
4. SELECT country_name, region_id, region_name
FROM regions JOIN countries USING (region_id)
WHERE region_id = 2;
5. SELECT e.first_name, e.last_name, e.hire_date, job_id, j.job_title, j.max_salary
FROM employees e JOIN jobs j USING (job_id)
WHERE max_salary > 12000;
6. SELECT job_title, first_name, last_name, email
FROM employees e JOIN jobs j USING (job_id)
WHERE job_id = 'ST_CLERK';
7. SELECT e.employee_id, e.first_name, e.last_name, w.manager_id, w.first_name,
w.last_name
FROM employees e JOIN employees w
ON (e.employee_id = w.manager_id);
8. SELECT l.location_id, l.city, d.department_name
FROM departments d JOIN locations l
ON (d.location_id = l.location_id)
WHERE l.country_id = 'CA';

```
9. SELECT e.manager_id,d.department_id, d.department_name, e.first_name,  
e.last_name  
FROM employees e JOIN departments d  
ON (e.department_id =d.department_id)  
WHERE d.department_id IN (80, 90, 110, 190);
```

```
10. SELECT e.employee_id, e.last_name, d.department_id, d.department_name,  
e.hire_date  
FROM employees e JOIN departments d  
ON (e.department_id = d.department_id)  
WHERE hire_date = '07-JUN-94';
```

11. a cross-join
 b Natural Join or Join..Using or Join...On
 c Join...On

Inner versus Outer Joins S04 L04

Vocabulary

<u>FULL OUTER JOIN</u>	Performs a join on two tables, retrieves all the rows in the Left table, even if there is no match in the Right table. It also retrieves all the rows in the Right table, even if there is no match in the Left table.
<u>Outer join</u>	A join that returns the unmatched rows as well as matched rows
<u>LEFT OUTER JOIN</u>	Performs a join on two tables, retrieves all the rows in the Left table even if there is no match in the Right table.
<u>RIGHT OUTER JOIN</u>	Performs a join on two tables, retrieves all the rows in the Right table even if there is no match in the Left table.
<u>Inner join</u>	A join of two or more tables that return only matched rows

Try It / Solve It

1. SELECT e.first_name, e.last_name, d.department_id
FROM employees e
LEFT OUTER JOIN departments d
ON(e.department_id = d.department_id);

2. SELECT e.first_name, e.last_name, d.department_id
FROM employees e
RIGHT OUTER JOIN departments d
ON(e.department_id = d.department_id);

3. SELECT e.first_name, e.last_name, d.department_id
FROM employees e
FULL OUTER JOIN departments d
ON(e.department_id = d.department_id);

4. SELECT w.last_name as "Employee", w.employee_id as "Emp#", m.last_name as
"Manager", m.employee_id as "Mgr#"
FROM employees w JOIN employees m
ON (w. manager_id = m.employee_id);

```
5. SELECT w.last_name as "Employee", w.employee_id as "Emp#", m.last_name as  
"Manager", m.employee_id as "Mgr#"  
FROM employees w  
LEFT OUTER JOIN employees m  
ON (w.manager_id = m.employee_id)  
ORDER BY w.employee_id;
```

```
6. SELECT c.first_name, c.last_name, e.event_date, e.description  
FROM d_clients c LEFT OUTER JOIN d_events e  
ON(c.client_number = e.client_number);
```

```
7. SELECT s.description, a.shift_assign_date  
FROM f_shifts s LEFT OUTER JOIN f_shift_assignments a ON (s.code = a.code);
```