

Oracle Academy

Database Programming with SQL

Instructor Resource Guide

Notes From PowerPoint Slides:

SECTION 13 LESSON 1 – TESTING

Slide 1: TESTING

No instructor notes for this slide

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 – UNIT TESTING

No instructor notes for this slide

Slide 5: Tell Me / Show Me – WHAT COULD BE TESTED?

No instructor notes for this slide

Slide 6: Tell Me / Show Me – WHAT SHOULD BE TESTED?

No instructor notes for this slide

Slide 7: Tell Me / Show Me – DESIGNING TESTS

No instructor notes for this slide

Slide 8: Tell Me / Show Me – RUNNING TESTS

No instructor notes for this slide

Slide 9: Tell Me / Show Me – Click to add text

No instructor notes for this slide

Slide 10: Summary – In this lesson you have learned to:

No instructor notes for this slide

SECTION 13 LESSON 2 – Final Project

Slide 1: Final Project

No instructor notes for this slide

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 – SCOPE OF THE FINAL PROJECT

No instructor notes for this slide

Slide 5: Tell Me / Show Me – EVALUATION OF THE FINAL PROJECT

Explain the Final Project Rubric to the students. Modify the rubric for your own class, if you'd like. The rubric is a guide, but it is meant to assess each student's application of SQL. Presentation skills should be a part of the final grade. The rubric can be found as a link in Section 0 - Instructor Resources.

Slide 6: Summary – In this lesson you have learned to:

No instructor notes for this slide

SECTION 13 LESSON 3 – Final Exam Review

Slide 1: Final Exam Review

No instructor notes for this slide

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: Why Learn It? (continued)

No instructor notes for this slide

Slide 5: Tell Me / Show Me – This is a review of the syntax

No instructor notes for this slide

Slide 6: Tell Me / Show Me – Case and Character Manipulation

No instructor notes for this slide

Slide 7: Tell Me / Show Me – Case and Character Manipulation (continued)

No instructor notes for this slide

Slide 8: Tell Me / Show Me – Number Functions

No instructor notes for this slide

Slide 9: Tell Me / Show Me – Date Functions

No instructor notes for this slide

Slide 10: Tell Me / Show Me – Conversion Functions

No instructor notes for this slide

Slide 11: Tell Me / Show Me – NULL Functions

No instructor notes for this slide

Slide 12: Tell Me / Show Me – Conditional Expressions

No instructor notes for this slide

Slide 13: Tell Me / Show Me – Cartesian Product and Join Operations

No instructor notes for this slide

Slide 14: Tell Me / Show Me – Nonequi Joins, Outer Joins

No instructor notes for this slide

Slide 15: Tell Me / Show Me – Nonequi Joins, Outer Joins (continued)

No instructor notes for this slide

Slide 16: Tell Me / Show Me – ANSI SQL Standard

No instructor notes for this slide

Slide 17: Tell Me / Show Me – ANSI SQL Standard (continued)

No instructor notes for this slide

Slide 18: Tell Me / Show Me – ANSI SQL Standard (continued)

No instructor notes for this slide

Slide 19: Tell Me / Show Me – ANSI SQL Standard (continued)

No instructor notes for this slide

Slide 20: Tell Me / Show Me – Group Functions, Group By Syntax

No instructor notes for this slide

Slide 21: Tell Me / Show Me – Single-row and multiple row Subqueries

No instructor notes for this slide

Slide 22: Tell Me / Show Me – Inserting, Updating and Deleting Data

No instructor notes for this slide

Slide 23: Tell Me / Show Me – Default Values

No instructor notes for this slide

Slide 24: Tell Me / Show Me – The Merge Statement

No instructor notes for this slide

Slide 25: Tell Me / Show Me – Creating Tables

No instructor notes for this slide

Slide 26: Tell Me / Show Me – Specifying Data Types

No instructor notes for this slide

Slide 27: Tell Me / Show Me – Modifying a table

No instructor notes for this slide

Slide 28: Tell Me / Show Me – Column Level Constraints

No instructor notes for this slide

Slide 29: Tell Me / Show Me – Table Level Constraints

No instructor notes for this slide

Slide 30: Tell Me / Show Me – Creating and Managing Views

No instructor notes for this slide

Slide 31: Tell Me / Show Me – Inline Views

No instructor notes for this slide

Slide 32: Tell Me / Show Me – Creating Sequences, Indexes, and Synonyms

No instructor notes for this slide

Slide 33: Tell Me / Show Me – Creating and Revoking Object Privileges

No instructor notes for this slide

Slide 34: Summary – In this lesson you have reviewed:

Have students take the Final Exam Practice Exam in Oracle iLearning. Review the database programming study guides and vocabulary quizzes. Complete the final exam prep questions for the SQL topics covered in Database Fundamentals.

Slide 35: Summary – In this lesson you have reviewed: (continued)

No instructor notes for this slide

Slide 36: Summary - Practice Guide

No instructor notes for this slide

Notes For Practice Activities:

Testing S13 L01

Try It / Solve It

1.

Test Number	Date	Test Description	Input	Expected Output	Result/Discrepancy	Action
1		a	INSERT INTO employees (employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, department_id) VALUES (300, 'Fred', 'Smith', 'a@b.com', 123456, '01 JAN 06', NULL,2000,NULL,100,90) ;	cannot insert NULL into JOB_ID	cannot insert NULL into JOB_ID	None
2		b	INSERT INTO job_history (employee_id, start_date, end_date, job_id, department_id) VALUES (400, '01 JAN 06', '01 DEC 05','IT_PROG',60) ;	Check constraint violated	Check constraint violated	None
3		c	DELETE departments WHERE department_id =10 ;	1 row deleted	integrity constraint violated - child record found	Alter referential integrity constraint
4		d	INSERT INTO employees (employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, department_id) VALUES (300, 'Fred', 'Smith', 'a@b.com', 123456, '01 JAN 06','IT_PROG',999,NULL, 100, 90) ;	Check constraint violated	1 row inserted	Alter check constraint

2

c. ALTER TABLE employees
DROP CONSTRAINT emp_dept_fk ;

ALTER TABLE employees
ADD CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)
REFERENCES departments(department_id) ON DELETE SET NULL ;

d. ALTER TABLE employees
DROP CONSTRAINT emp_salary_min ;

ALTER TABLE employees
ADD CONSTRAINT emp_salary_min CHECK (salary >= 1000) ;

Final Project S13 L02

Try It / Solve It

No solutions provided.

Final Exam Review S13 L03

Try It / Solve It

Part 1

1. SELECT first_name, last_name, zip
FROM f_customers
WHERE LENGTH(zip) < 10;

- 2. a. LPAD (CDN)
- b. ROUND (DN)
- c. TRUNC (DN)
- d. LENGTH (C)
- e. LAST_DAY (D)
- f. INSTR (CDN)
- g. CONCAT (CDN)

3. SELECT first_name, last_name, NVL(TO_CHAR(auth_expense_amt, '99999999'), 'Not Approved') As "Authorization Status"
FROM d_partners;

4. Jason and Jamie forgot that the data types of the NVL elements must be the same data type! A TO_CHAR function is needed to convert the number overtime_rate to a character value.

SELECT first_name, last_name, NVL(TO_CHAR(overtime_rate, '99999999'), 'no overtime') As "Payrate"
FROM f_staffs;

5. SELECT first_name, last_name, birthdate, SUBSTR(TO_CHAR(birthdate,'Month dd,YYYY'), 1,9)|| ' 2005' As "Send Card"
FROM f_staffs;

6. ___T___a. TO_CHAR is required to convert the date '03-JUN-04' to June 3, 2004
___F___b. TO_NUMBER will convert '23-NOV-02' to use with ADD_MONTHS
___F___c. TO_DATE will convert SYSDATE to today's date
___F___d. TO_NUMBER('101', '\$99999') will convert 101 to a number
___T___e. TO_CHAR(salary, '\$9999.99') will convert number to character format
___T___f. TO_NUM(varchar2 column) will convert character data to a number
___T___g. TO_CHAR(SYSDATE, 'Month fmdd, yyyy') will format the date

7. SELECT first_name,
last_name,ROUND(MONTHS_BETWEEN(SYSDATE,hire_date)/12 ,1)AS "Years
Worked"
FROM employees;

8. SELECT SUBSTR(address, -5,5)AS "Old Zip", SUBSTR(address, -5,5)||'-2345' AS
"New Zip"
FROM d_venues
WHERE ID= 105;

9. SELECT ROUND(SYSDATE, 'YEAR') AS YEAR, ROUND(SYSDATE,
'MONTH')AS MONTH, TRUNC(SYSDATE, 'YEAR')AS YEAR, TRUNC(SYSDATE,
'MONTH')AS MONTH
FROM DUAL;

10. SELECT name, ROUND((SYSDATE - START_DATE),0) AS "Days"
FROM f_promotional_menus;

11. SELECT LOWER(CONCAT(LOWER(SUBSTR(job_title, 1, 5)),'* ')) "Job
Description"
FROM jobs;

12. The expression will be evaluated from the innermost expression outward.

Part 2

These are just a few example queries. Students should run their queries against the database to check their accuracy.

1. SELECT employee_id, last_name, department_id
FROM employees
WHERE LOWER(last_name) = 'ernst';

SELECT employee_id, CONCAT(first_name, last_name) NAME,
job_id, LENGTH (last_name),
INSTR(last_name, 'i') "Contains 'i'?"
FROM employees
WHERE SUBSTR(job_id, 4) = 'PRES';

SELECT ROUND(45.923,2), ROUND(45.923,0),
ROUND(45.923,-1)
FROM DUAL;


```
2. SELECT UPPER(CONCAT(cd_number, title))
FROM d_cds WHERE cd_number = 94;
SELECT UPPER(cd_number)||UPPER(title)
FROM d_cds WHERE cd_number = 94;
```

3. ___F___a. LOWER converts numbers to lowercase.
 ___F___b. Use RPAD to move numbers to the right to place an * on the left.
 ___F___c. TRIM can be used to trim one or more characters from a string.
 ___T___d. LENGTH returns a number.
 ___F___e. SUBSTR is used to substitute one string for another.
 ___T___f. CONCAT is limited to using two parameters.
 ___T___g. TRUNC will return zero decimal places if a decimal value is omitted.

```
4. SELECT TO_CHAR(cost, '$99999.99')
FROM d_events;
```

```
5. SELECT '*'||ID AS "New Id"
FROM f_staffs WHERE LENGTH(ID) = 1;
```

```
6. SELECT TO_DATE('December 15, 1995', 'Month dd, RRRR')
FROM DUAL;
```

```
7. SELECT LOWER(TO_CHAR(TO_DATE('19-JUN-04','DD-MON-YY'), 'ddth "of"
Month YYYYsp'))
FROM DUAL ;
```

```
8. SELECT SUBSTR('Oracle Academy', -7,7)
FROM DUAL;
```

9. There is nothing wrong with the syntax; there is no matching data in the database.

10. _C___a. To convert varchar2 to number data
 _C___b. To format a date to other than the default format
 _C___c. To convert a date such as June 19, 2000 to default format
 _N___d. To format a number to appear as currency
 _G___e. To substitute a value in a table for null
 _CE__f. To do an IF-THEN-ELSE statement
 _G___g. To find the first not null expression among a list of expressions
 _C___h. To replace a section of a string with another string
 _C___i. To format a 20th-century date
 _C___j. To present output all in uppercase
 _C___k. To find the numeric position of a character in a string
 _D___l. To find the last day of the month

Part 3

1. Cartesian product or cross join
2. SELF JOIN
3. OUTER JOIN
4. Deficient or is missing
5. Nonequijoin
6. WHERE
7. table alias
8. FROM
9. will
10. 30
11. +
12. LEFT OUTER JOIN
13. Self joins
14. 2
15. Oracle syntax that has been developed by the company
16. Join the tables with the USING clause.
17. Equijoin and natural join
18. Cartesian product
19. What - SELECT
Where - FROM
How - WHERE
20.
_T___cross-join
_T__ equijoin
_T___nonequijoin
____natural join
T___full outer join
T___left outer join
____USING clause
21. a. Joins with the USING clause
b. Natural join when the columns have the same name
22. Cross join or Cartesian product
23. a. EQUIJOIN
SELECT s.name, c.name
FROM students s, course c, enrolled e
WHERE (s.sid = e.sid) AND (e.cid = c.cid);
b. FROM students NATURAL JOIN enrolled or
FROM courses NATURAL JOIN enrolled;
c. SELF JOIN
SELECT s.name, m.name
FROM students s, students m
WHERE s.mid=m.sid;
d. SELECT s.name, m.name
FROM students s, students m
WHERE s.mid=m.sid(+);
SELECT s.name, m.name

```
FROM students s LEFT OUTER JOIN students m
ON (s.mid=m.sid);
```

Part 4

1. Create the three o_tables, jobs, employees and departments using the syntax:

```
CREATE TABLE o_jobs
AS (SELECT * FROM jobs);
```

2. Add the Human Resources job to the jobs table:

```
INSERT INTO o_jobs (job_id, job_title, min_salary, max_salary)
VALUES('HR_MAN', 'Human Resources Manager', 4500, 5500);
```

Don't accept this assignment until the correct output is produced. Students should have added three employees to the o_employees table, added the 'HR_MAN' job to the o_jobs table, and added the 'Human Resources' department to the o_departments table. Verify that students were successful in producing the correct output (ask them to print or show you their output)

Changes to these tables will be done in Lesson 3.

3.

```
INSERT INTO o_employees (employee_id, first_name, last_name, email, hire_date,
job_id)
```

```
VALUES(210, 'Ramon', 'Sanchez', 'RSANCHEZ', SYSDATE, 'HR_MAN');
```

```
INSERT INTO o_employees (employee_id, first_name, last_name, email, hire_date,
job_id)
```

```
VALUES(211, 'Tai', 'Sugita', 'TSUGITA', SYSDATE, 'HR_MAN');
```

```
INSERT INTO o_employees (employee_id, first_name, last_name, email, hire_date,
job_id)
```

```
VALUES(212, 'Alina', 'Arcos', 'AARCOS', SYSDATE, 'HR_MAN');
```

4.

```
INSERT INTO o_departments(department_id, department_name)
```

```
VALUES (210,'Human Resources');
```

Don't accept this assignment until the correct output is produced. Students should have added three employees to the o_employees table, added the 'HR_MAN' job to the o_jobs table, and added the 'Human Resources' department to the o_departments table. Verify that students were successful in producing the correct output (ask them to print or show you their output).

Changes to these tables will be done in Lesson 3.

5.

```
UPDATE o_employees
```

```
SET phone_number = '360.509.7132'
```

```
WHERE employee_id = 210;
```

```
UPDATE o_departments
```

```
SET location_id = 1700
```

```
WHERE department_id= 210;
```

```
UPDATE o_employees
```

```
SET department_id = 210
WHERE employee_id in (210,211,212);
```

```
UPDATE o_employees
SET salary = 5000
WHERE employee_id = 210;
```

```
UPDATE o_employees
SET salary = 5100
WHERE employee_id = 211;
```

```
DELETE FROM o_employees
WHERE employee_id = 212;
```

Additional information: Verify that the new changes have been made to the o_tables as specified. Explain that DML statements should be made to primary-key values such as employee ID instead of last name or first name just in case two employees have the same name. Two employees won't have the same ID.

For example, in a large company, two employees may be named Jennifer James, but each employee will have a different employee identification number. When changes are made to Jennifer James's email address, it is best to use the query shown on the left in the following example rather than the query shown on the right.

```
UPDATE o_employees
SET email = 'jjames@charter.net'
WHERE employee_id = 210;
```

```
UPDATE o_employees
SET email = 'jjames@charter.net'
WHERE last_name = 'James';
```

Part 5

1. Constraints enforce rules and prevent deletion of tables when there are dependencies.
2. CHECK constraints specify a condition that must be True. Example: A negative salary amount cannot be entered in a table.
3. Query the data dictionary USER_CONSTRAINTS.
4. When a table is created or added after the table is created (depending on data already in the table).
5. The SYSn name does not explicitly name which column the constraint refers to.
6. Referential integrity prevents additions, deletions, or modifications of information that violates business rules.
7. It is really a personal preference, but NON NULL constraints must be created at the column level and composite keys at the table level.
8. To prevent duplicate information in a column that is not a primary-key column
9. Yes, because nulls are not equal to anything -- just can't have identical values in nonnull columns

10. last name and email address
11. one
12. nonduplicate values in a column
13. referential integrity constraint
14. ODC deletes the corresponding foreign key referenced row when parent rows are deleted
15. Release the foreign key constraint on the parent table.

Part 6

1. b. The value 999 will not be generated by this sequence
 - f. USER_OBJECTS documents this sequence in the data dictionary.
 2. d. email VARCHAR2(25) CONSTRAINT emp_email_uk UNIQUE,
 - h. CONSTRAINT employee_id_pk PRIMARY KEY(employee_id),
 3. b. USER_SEQUENCES
 4. e. None of the above
- Indexes cannot be modified -- instead, DROP and re-create.
5. c. USER_IND_COLUMNS
 6. c. The synonym is not accessible to all users.
 7. b. 6, 9, 12
 8. c. Most queries are expected to return more than 5% of the rows.
 9. b. FOREIGN KEY
 - c. PRIMARY KEY
 - d. UNIQUE COMPOSITE KEY
 10. b. FROM

