

Oracle Academy

Database Programming with SQL

Instructor Resource Guide

Notes For PowerPoint Slides:

SECTION 9 LESSON 1 – Defining NOT NULL and UNIQUE Constraints

Slide 1: Defining NOT NULL and UNIQUE Constraints

Lesson Preparation

Giving constraints meaningful names is important for later documentation. Enforce and practice the naming convention that applies to all types of constraints: table name_column name_constraint type. For example, a primary-key constraint on the employee_id column of the employees table would be named: emp_emp_id_pk.

What to Watch For

Students may have trouble distinguishing between the constraint definitions at the column or table level.

Defining constraints will create many syntax errors if not coded carefully. Show students the details of the CREATE TABLE statement: commas end each column definition; parentheses enclose the column definitions; constraint names incorporate underscores if they are made up of more than one word. For example: g_loc_address_pk.

Connections

Practice the CREATE TABLE statement with entity diagrams from data modeling such as the ERD from the Animal Shelter activity.

Slide 2: What Will I Learn?

No instructor notes for this slide

Slide 3: Why Learn It?

Share with students other constraints we live with in everyday life. Ask students for their own constraints! Example: you can't have the same student identification number as someone else.

Slide 4: Tell Me / Show Me – Constraints in General

Students have already come the NOT NULL constraint although they have not yet learnt how to create it. Use NOT NULL as an introductory example of a strict database rule.

Stress that no user can break a constraint rule. Of course the DBA has privileges to drop any constraint; but as long as the constraint exists, not even the DBA can violate it.

Slide 5: Tell Me / Show Me – CREATING CONSTRAINTS

Review the CREATE TABLE syntax with students. Point out that we cannot tell which columns are the primary or foreign-key columns.

This and the next lesson show how to specify constraints as part of CREATE TABLE. The third lesson will show how to add constraints later, after the table has been created.

Slide 6: Tell Me / Show Me – Constraints at the Column Level

Point out that this constraint definition is part of the column definition, just like the column data type. Where is the comma which separates this column definition from the next one? After the constraint definition.

Slide 7: Tell Me / Show Me – Naming Constraints

No instructor notes for this slide

Slide 8: Tell Me / Show Me – Naming Constraints at the Column Level

No instructor notes for this slide

Slide 9: Tell Me / Show Me – This example shows both a user-named...

No instructor notes for this slide

Slide 10: Tell Me / Show Me – CONSTRAINTS AT THE TABLE LEVEL

Point out the comma that ends the column definitions. The table-level constraints are then listed below ending with);

Slide 11: Tell Me / Show Me – Basic Rules For Constraints

Explain why these rules are necessary:

- a constraint which refers to two columns cannot be defined as part of a column definition. Which of the two column definitions would we put it alongside ? It couldn't be both!

- a table level constraint must be user-named, because table level constraints must include the keyword CONSTRAINT

It is not obvious why NOT NULL constraints cannot be defined at the table level. The reason is: because the ANSI/ISO SQL standard forbids it.

Slide 12: Tell Me / Show Me – Examine the Violations

Review the Violators Beware graphic. Point out that if two or more table level constraints are defined, they are separated by commas.

Ask students to correct the example.

Answers will vary:

```
CREATE TABLE clients (  
  client_number    NUMBER(6),  
  first_name       VARCHAR2(20),  
  last_name        VARCHAR2(20),  
  phone            VARCHAR2(20),  
  email            VARCHAR2(10) CONSTRAINT clients_email_nn NOT NULL,  
  CONSTRAINT clients_phone_email_uk UNIQUE(email,phone),  
  CONSTRAINT clients_client_num_pk PRIMARY KEY (client_number));
```

Slide 13: Tell Me / Show Me – FIVE TYPES OF CONSTRAINT

No instructor notes for this slide

Slide 14: Tell Me / Show Me – NOT NULL CONSTRAINT

Remind students that NOT NULL means that a value must be assigned. Any value that works with the data type defined for the column can be entered. A zero or a letter or a number or even a space are considered "values." A NULL means "no value."

Slide 15: Tell Me / Show Me – UNIQUE CONSTRAINT

No instructor notes for this slide

Slide 16: Tell Me / Show Me – An example of UNIQUE:

Use the table graphic to show the email columns. If two people lived in the same household and shared an email address, an email address could not be shared in this table. Each email address must be unique.

Slide 17: Tell Me / Show Me – When defining UNIQUE constraints, it is...

Ask students to give examples of composite unique-key columns.

Answers will vary: city and state, last_name and email, student_id and last_name

Slide 18: Tell Me / Show Me – UNIQUE constraints allow the input ...

A UNIQUE constraint enforces the rule that all values must be unique. NULL is not a value. You could create a table with a UNIQUE column, then INSERT 1000 rows into the table, all with a null in the column.

Slide 19: Tell Me / Show Me – Terminology

UNIQUE constraint-every value in a column or set of columns (a composite key) be unique

NOT NULL constraint-for every row entered into the table, there must be a value for that column

PRIMARY KEY-constraint ensures that the column contains no null values and uniquely identifies each row of the table

CHECK constraint-specifies a condition that must be true for each row of data

REFERENCES-identifies that table and column in the parent table

UNIQUE KEY-an integrity constraint that requires every value in a column or set of columns be unique

FOREIGN KEY-designates a column (child table) that establishes a relationship between a primary key in the same table and a different table (parent table)

Table level constraint-references one or more columns and is defined separately from the definitions of the columns in the table

Constraint-Database rules.

Column- level constraint-Database rule that references a single column

Slide 20: Summary – In this lesson you have learned to:

No instructor notes for this slide

Slide 21: Summary - Practice Guide

No instructor notes for this slide

SECTION 9 LESSON 2 – PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

Slide 1: PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

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 –

Begin this lesson with a review question. When creating a table, how do you specify a column to have a NOT NULL and UNIQUE KEY constraint?

client_number NUMBER(7) NOT NULL

client_number NUMBER(7) CONSTRAINT clients_client_num_d_clients_nn NOT NULL

Slide 5: Tell Me / Show Me – PRIMARY KEY constraints can be ...

No instructor notes for this slide

Slide 6: Tell Me / Show Me – In a CREATE TABLE statement, the ...

No instructor notes for this slide

Slide 7: Tell Me / Show Me – To define a composite PRIMARY KEY, ...

No instructor notes for this slide

Slide 8: Tell Me / Show Me – FOREIGN KEY (REFERENTIAL INTEGRITY)

No instructor notes for this slide

Slide 9: Tell Me / Show Me – To state a FOREIGN KEY ...

No instructor notes for this slide

Slide 10: Tell Me / Show Me – The table containing the foreign key is called...

No instructor notes for this slide

Slide 11: Tell Me / Show Me – To satisfy a referential-integrity constraint...

No instructor notes for this slide

Slide 12: Tell Me / Show Me – The rule is: before you define a...

No instructor notes for this slide

Slide 13: Tell Me / Show Me – To define a FOREIGN KEY

No instructor notes for this slide

Slide 14: Tell Me / Show Me – The syntax for defining a FOREIGN KEY...
No instructor notes for this slide

Slide 15: Tell Me / Show Me – ON DELETE CASCADE
No instructor notes for this slide

Slide 16 Tell Me / Show Me – If the song_id column in D_TRACK_LISTING...
No instructor notes for this slide

Slide 17: Tell Me / Show Me – Column-Level ON DELETE CASCADE syntax:
No instructor notes for this slide

Slide 18: Tell Me / Show Me – ON DELETE SET NULL
No instructor notes for this slide

Slide 19: Tell Me / Show Me – CHECK CONSTRAINTS
No instructor notes for this slide

Slide 20: Tell Me / Show Me – CREATE d_cds (cd_number
No instructor notes for this slide

Slide 21: Tell Me / Show Me – CHECK Constraint Conditions
No instructor notes for this slide

Slide 22: Tell Me /Show Me – CHECK constraints can be defined
No instructor notes for this slide

Slide 23 Tell Me / Show Me – Terminology

ON DELETE CASCADE-Allows a foreign key row that is referenced to a primary key row to be deleted

CHECK constraint-explicitly defines a condition that must be met

PRIMARY KEY constraint-a column or set of columns that uniquely identifies each row in a table

NOT NULL-constraint ensures that the column contains no null values

ON DELETE SET NULL-Allows a row to stay in a table when the data is deleted without deleting the whole row

FOREIGN KEY constraint-establishes a relationship between the foreign key column and a primary key or unique key in the same table or a different table

Slide 24: Summary – In this lesson you have learned to:
No instructor notes for this slide

Slide 25: Summary - Practice Guide
No instructor notes for this slide

SECTION 9 LESSON 3 – Managing Constraints

Slide 1: Managing Constraints

Connections

Relate the managing of constraints to the work students completed in making ERDs in data modeling. Making sure the correct relationships are identified for all tables is important during the design process. If the correct relationships are not established in the design stage, it may be difficult or impossible to add constraints after data has been entered in the tables. Deleting data to add constraints would not be desirable.

Slide 2: What Will I Learn?

No instructor notes for this slide

Slide 3: Why Learn It?

Slide 4: Why Learn It? (continued)

No instructor notes for this slide

Slide 5: Tell Me / Show Me – MANAGING CONSTRAINTS:

No instructor notes for this slide

Slide 6: Tell Me / Show Me – The ALTER statement requires:

No instructor notes for this slide

Slide 7: Tell Me / Show Me – ADDING CONSTRAINTS

To add a constraint to an existing table, use the ...

No instructor notes for this slide

Slide 8: Tell Me / Show Me – ADDING CONSTRAINTS

If the constraint is a FOREIGN KEY constraint...

No instructor notes for this slide

Slide 9: Tell Me / Show Me – Consider the DJs on Demand database

Explain the syntax in the example shown. Ask students why ON DELETE CASCADE was added to the foreign-key constraint. Answer: to enable the parent (D_CLIENT) row to be deleted if there are matching rows in the child (D_EVENTS) table.

Slide 10: Tell Me / Show Me – If the constraint is NOT NULL

No instructor notes for this slide

Slide 11: Tell Me / Show Me – WHY ENABLE AND DISABLE CONSTRAINTS?

Constraint checking, although done automatically, takes time, especially checking foreign key constraints. DBAs sometimes speed up high-volume batch operations by disabling constraints. The risk is that invalid changes could be made to a table and not be identified until later, when the constraint is re-enabled.

Slide 12: Tell Me / Show Me – DROPPING CONSTRAINTS

To drop the primary-key constraint on the D_CLIENTS table, the word CASCADE is used to drop the foreign- key constraint in the child table

```
ALTER TABLE c_clients DROP PRIMARY KEY CASCADE;
```

Note: Since there is only one primary key for a table, we did not have to specify the column name

Slide 13: Tell Me / Show Me – DISABLING CONSTRAINTS

No instructor notes for this slide

Slide 14: Tell Me / Show Me – You can use the DISABLE clause...

Explain the syntax in the examples shown. Ask students to give an example where it might be necessary to disable a constraint in the D_CLIENTS table. Event data needs to be entered in the child D_EVENTS table, but the client_number is not known as yet. Use the SELECT statement syntax to view USER_CONSTRAINTS and USER_CONS_COLUMNS. Although creating indices is not included in this lesson, students may ask what they are. Indices are created in an existing table to locate rows more quickly and efficiently. It is possible to create an index on one or more columns of a table, and each index is given a name. The users cannot see the indexes; they are just used to speed up queries.

Slide 15: Tell Me / Show Me – The CASCADE clause

No instructor notes for this slide

Slide 16: Tell Me / Show Me – ENABLING CONSTRAINTS

No instructor notes for this slide

Slide 17: Tell Me / Show Me – If you enable a constraint, that...

No instructor notes for this slide

Slide 18: Tell Me / Show Me – CASCADING CONSTRAINTS:

No instructor notes for this slide

Slide 19: Tell Me / Show Me – If all columns referenced by the...

No instructor notes for this slide

Slide 20: Tell Me / Show Me – VIEWING CONSTRAINTS

No instructor notes for this slide

Slide 21: Tell Me / Show Me – To view all constraints on your ...

No instructor notes for this slide

Slide 22: Tell Me / Show Me – VIEWING CONSTRAINTS

Students need to know how to use the SELECT statement to view the data dictionary constraints. Make sure they write this code so that they will be familiar with it.

```
SELECT constraint_name, column_name  
FROM user_cons_columns  
WHERE table_name = 'EMPLOYEES';
```

Slide 23: Tell Me /Show Me - Terminology

DISABLE CONSTRAINT-to deactivate an integrity constraint

CASCADE clause-disables dependent integrity constraints

ALTER TABLE-to add, modify, or drop columns from a table

ENABLE CONSTRAINT-to activate an integrity constraint currently disabled

DROP CONSTRAINT-removes a constraint from a table

DROP COLUMN-Allows user to delete a column from a table

CASCADE RESTRAINTS clause-defines the actions the database server takes when a user attempts to delete or update a key to which existing foreign keys point

Slide 24: Summary – In this lesson you have learned to:

No instructor notes for this slide

Slide 25: Summary - Practice Guide

No instructor notes for this slide

Notes For Practice Activities:

Defining NOT NULL and UNIQUE Constraints S09 L01

Vocabulary

<u>UNIQUE constraint</u>	Every value in a column or set of columns (a composite key) be unique
<u>NOT NULL constraint</u>	For every row entered into the table, there must be a value for that column
<u>PRIMARY KEY</u>	Constraint ensures that the column contains no null values and uniquely identifies each row of the table
<u>CHECK constraint</u>	Specifies a condition that must be true for each row of data
<u>REFERENCES</u>	Identifies that table and column in the parent table
<u>UNIQUE KEY</u>	An integrity constraint that requires every value in a column or set of columns be unique
<u>FOREIGN KEY</u>	Designates a column (child table) that establishes a relationship between a primary key in the same table and a different table (parent table)
<u>Table level constraint</u>	References one or more columns and is defined separately from the definitions of the columns in the table
<u>Constraint</u>	Database rules.
<u>Column- level constraint</u>	Database rule that references a single column

Try It / Solve It

1. A constraint is a database rule
2. A column level constraint applies to only one column, consequently composite keys may not be applied at the column level. The NOT NULL constraint may only be applied at the column level.
3. The Oracle server will automatically assign a meaningless name to constraint if it is not given a meaningful one making it difficult to identify a constraint from its name.
4. See table in student guide
5. See table in student guide.

```
6. CREATE TABLE global_locations (  
  Id          NUMBER(4) CONSTRAINT g_loc_id_nn NOT NULL,  
  loc_name    VARCHAR2(20),  
  date_opened DATE CONSTRAINT g_loc_date_nn NOT NULL,  
  address     VARCHAR2(30) CONSTRAINT g_loc_address_nn NOT NULL,  
  city        VARCHAR2(20) CONSTRAINT g_loc_city_nn NOT NULL,  
  zip_postal_code VARCHAR2(20),  
  phone       VARCHAR2(15),  
  email       VARCHAR2(15) CONSTRAINT g_loc_email_uk UNIQUE,  
  manager_id  NUMBER(4),  
  emergency_contact VARCHAR2(20));
```

7. Execute the CREATE TABLE statement in HTML DB.

8. DESCRIBE global_locations;

```
9. CREATE TABLE global_locations (  
  Id          NUMBER(4) CONSTRAINT g_loc_id_nn NOT NULL,  
  loc_name    VARCHAR2(20),  
  date_opened DATE CONSTRAINT g_loc_date_nn NOT NULL,  
  address     VARCHAR2(30) CONSTRAINT g_loc_address_nn NOT NULL,  
  city        VARCHAR2(20) CONSTRAINT g_loc_city_nn NOT NULL,  
  zip_postal_code VARCHAR2(20),  
  phone       VARCHAR2(15),  
  email       VARCHAR2(15),  
  manager_id  NUMBER(4),  
  emergency_contact VARCHAR2(20),  
  CONSTRAINT g_loc_email_uk UNIQUE(email));
```

PRIMARY KEY, FOREIGN KEY, and CHECK Constraints S09 L02

Vocabulary

<u>ON DELETE CASCADE</u>	Allows a foreign key row that is referenced to a primary key row to be deleted
<u>CHECK constraint</u>	Explicitly defines a condition that must be met
<u>PRIMARY KEY constraint</u>	A column or set of columns that uniquely identifies each row in a table
<u>NOT NULL</u>	Constraint ensures that the column contains no null values
<u>ON DELETE SET NULL</u>	Allows a row to stay in a table when the data is deleted without deleting the whole row
<u>FOREIGN KEY constraint</u>	Establishes a relationship between the foreign key column and a primary key or unique key in the same table or a different table

Try It / Solve It

1. To prevent mistakes, students should write this out on paper or in a word-processing program before creating the tables.

```
animal_id          NUMBER(6)
name               VARCHAR2(25)
license_tag_number NUMBER(10)
admit_date         DATE
adoption_id        NUMBER(5),
vaccination_date   DATE
```

```
2. CREATE TABLE animals (
  animal_id          NUMBER(6),
  name               VARCHAR2(25),
  license_tag_number NUMBER(10),
  admit_date         DATE CONSTRAINT animals_admit_nn NOT NULL,
  adoption_id        NUMBER(5),
  vaccination_date   DATE CONSTRAINT animals_vacc_nn NOT NULL,
  CONSTRAINT animal_id_pk PRIMARY KEY (animal_id),
  CONSTRAINT lic_tag_num_uk UNIQUE (license_tag_number));
```

```
3. INSERT INTO animals (animal_id, name, license_tag_number, admit_date,
  adoption_id, vaccination_date)
```

VALUES (101, 'Spunky', 35540, '10-OCT-04', 205, '12-OCT-04');

4. Column level: adoption_id NUMBER(5) CONSTRAINT adopt_date_fk
REFERENCES adoptions(adoption_id);

Table level: CONSTRAINT adopt_date_fk FOREIGN KEY(adoption_id)
REFERENCES adoptions (adoption_id);

5a. When the parent record is removed so is the child record.

5b. When the parent record is removed then the foreign key field is set to null.

6.

- A CHECK constraint cannot be used in queries that refer to values in other rows, even in the same table
- The CHECK constraint cannot contain calls to the functions SYSDATE, UID, USER, or USERENV.
- The CHECK constraint cannot use the pseudocolumns CURRVAL, NEXTVAL, LEVEL, or ROWNUM..

Managing Constraints S09 L03

Vocabulary

<u>DISABLE CONSTRAINT</u>	To deactivate an integrity constraint
<u>CASCADE clause</u>	Disables dependent integrity constraints
<u>ALTER TABLE</u>	To add, modify, or drop columns from a table
<u>ENABLE CONSTRAINT</u>	To activate an integrity constraint currently disabled
<u>DROP CONSTRAINT</u>	Removes a constraint from a table
<u>DROP COLUMN</u>	Allows user to delete a column from a table
<u>CASCADE RESTRAINTS clause</u>	Defines the actions the database server takes when a user attempts to delete or update a key to which existing foreign keys point

Try It / Solve It

The practice exercises use the d_clients and d_events tables in the DJ on Demand database. Students will work with copies of these two tables named copy_d_clients and copy_d_events. Make sure they have new copies of the tables (without changes made from previous exercises). Remember, tables copied using a subquery do not have the

integrity constraints as established in the original tables. When using the SELECT statement to view the constraint name, the tablename must be all capital letters.

1. Add, drop enable, disable a constraint

2. ALTER TABLE copy_d_clients
ADD CONSTRAINT copy_d_clients_pk PRIMARY KEY(client_number);

3. ALTER TABLE copy_d_events
ADD CONSTRAINT copy_d_events_fk FOREIGN KEY(client_number)
REFERENCES copy_d_clients(client_number);

4a. SELECT constraint_name, constraint_type
FROM USER_CONSTRAINTS
WHERE table_name = 'COPY_D_CLIENTS';

CONSTRAINT_NAME	CONSTRAINT_TYPE
COPY_D_CLIENTS_PK	P
SYS_C00583282	C
SYS_C00583283	C
SYS_C00583284	C

4b. SELECT constraint_name, constraint_type
FROM USER_CONSTRAINTS
WHERE table_name = 'COPY_D_EVENTS';

CONSTRAINT_NAME	CONSTRAINT_TYPE
COPY_D_EVENTS_FK	R
SYS_C00583275	C
SYS_C00583276	C
SYS_C00583277	C
SYS_C00583278	C
SYS_C00583279	C
SYS_C00583280	C
SYS_C00583281	C

5. Answers will vary. The attempt to drop was unsuccessful.

6. INSERT INTO copy_d_events (id,
name,event_date,description,cost,venue_id,package_code,theme_code,client_number)
VALUES(140,'Cline Bas Mitzvah','15-JUL-04','Church and Private Home,
formal',4500,105,87,77,7125);
ORA-02291: integrity constraint (USTA_SDHS_SQL01_S01.COPY_D_EVENTS_FK)
violated - parent key not found

**The attempt to drop the primary-key column in the copy_d_clients table was not successful in step 5. Therefore, the attempt to insert a row that has a value in a foreign-key child column that does not have a corresponding value in a parent column will also fail.

```
6. ALTER TABLE copy_d_clients
DISABLE CONSTRAINT copy_d_clients_pk;
ORA-02297: cannot disable constraint
(USTA_SDHS_SQL01_S01.COPY_D_CLIENTS_PK) - dependencies exist
ALTER TABLE copy_d_clients
DISABLE CONSTRAINT copy_d_clients_pk CASCADE;
```

If the ALTER TABLE query did not have the CASCADE option, the FOREIGN KEY constraint was not disabled, so the ALTER TABLE DISABLE failed. If the CASCADE option was added to the ALTER TABLE query, the INSERT will be successful.

```
8. INSERT INTO copy_d_events (id,
name,event_date,description,cost,venue_id,package_code,theme_code,client_number)
VALUES(140,'Cline Bas Mitzvah','15-JUL-04','Church and Private Home,
formal',4500,105,87,77,7125);
```

The primary key has been disabled, allowing values to be inserted into the copy_d_events table even though there is no corresponding client_number in the parent table copy_d_clients.

```
9. ALTER TABLE copy_d_clients;
ENABLE CONSTRAINT copy_d_clients_pk;
```

The ENABLE was successful even though the new addition in the copy_d_events table has a client number not found in the parent table. When enabling a primary key that was disabled with the CASCADE option, the foreign keys are not enabled.

10. The copy_d_clients table would need to have the client number and information added to it before the copy_d_events table could reenable the foreign-key constraint on the client_number column.

11. In certain situations, however, it is desirable to temporarily disable the integrity constraints of a table temporarily for performance reasons, such as:

- When loading large amounts of data into a table
- When performing batch operations that make massive changes to a table (such as changing everyone's employee number by adding 1,000 to the existing number)

12. C stands for CHECK, P for PRIMARY KEY, R for REFERENTIAL INTEGRITY, and U for UNIQUE. NOT NULL is shown as a C because it is implemented as a CHECK CONSTRAINT.