

EXERCISE 12

Intro to Constraints; NOT NULL and UNIQUE Constraints

Global Fast Foods has been very successful this past year and has opened several new stores. They need to add a table to their database to store information about each of their store's locations. The owners want to make sure that all entries have an identification number, date opened, address, and city and that no other entry in the table can have the same email address. Based on this information, answer the following questions about the global_locations table. Use the table for your answers.

Global Fast Foods global_locations Table						
NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT
Id						
name						
date_opened						
address						
city						
zip/postal code						
phone						
email						
manager_id						
Emergency contact						

1. What is a "constraint" as it relates to data integrity?

A constraint is a rule that ensures accuracy and consistency of data in a table

2. What are the limitations of constraints that may be applied at the column level and at the table level?

column level : applies only to one column

Table level : can apply to multiple columns but not to computed values.

3. Why is it important to give meaningful names to constraints?

It helps identify and manage them easily when debugging or altering the table.

4. Based on the information provided by the owners, choose a datatype for each column. Indicate the length, precision, and scale for each NUMBER datatype.

Id - NUMBER(4) - NOT NULL - Primary key
manager_id - NUMBER(4) - NULL

5. Use "(nullable)" to indicate those columns that can have null values.

columns → date-opened, address, city, zip-postal, phone, manager_id, contact.

6. Write the CREATE TABLE statement for the Global Fast Foods locations table to define the constraints at the column level.

```
CREATE TABLE global_locations (
  id NUMBER(4) CONSTRAINT loc_id_nn NOT NULL,
  loc_name VARCHAR2(20) CONSTRAINT loc_name_nn
    NOT NULL,
  date_opened DATE,
  address VARCHAR2(30),
  city VARCHAR2(20),
  zip_postal VARCHAR2(20),
  phone VARCHAR2(15),
  email VARCHAR2(80) CONSTRAINT loc_email_uk
    UNIQUE NOT NULL,
  manager_id NUMBER(4),
  contact VARCHAR2(40)
);
```

7. Execute the CREATE TABLE statement in Oracle Application Express.

Run the above statement.

8. Execute a DESCRIBE command to view the Table Summary information.

```
DESC global_locations;
```

9. Rewrite the CREATE TABLE statement for the Global Fast Foods locations table to define the UNIQUE constraints at the table level. Do not execute this statement.

NAME	TYPE	LENGTH	PRECISION	SCALE	NULLABLE	DEFAULT
id	number	4				
loc_name	varchar2	20			X	
	date					
address	varchar2	30				
city	varchar2	20				
zip_postal	varchar2	20			X	
phone	varchar2	15			X	
email	varchar2	80			X	
manager_id	number	4			X	
contact	varchar2	40			X	

```
CREATE TABLE global_locations (
```

```
  id NUMBER(4) NOT NULL,
  loc_name VARCHAR(20) NOT NULL,
  date_opened DATE,
  address VARCHAR2(30),
  city VARCHAR(20),
  zip_postal VARCHAR(20),
  phone VARCHAR(15),
  email VARCHAR2(80) NOT NULL,
  manager_id NUMBER(4),
  contact VARCHAR2(40),
  CONSTRAINT loc_id_pk
    PRIMARY KEY (id),
  CONSTRAINT loc_email_uk
    UNIQUE (email)
);
```


PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

1. What is the purpose of a
 - PRIMARY KEY *uniquely identify each row, cannot be null*
 - FOREIGN KEY *enforces referential integrity by linking to another table's primary key*
 - CHECK CONSTRAINT *ensures column values meet a condition (eg: salary > 0)*

2. Using the column information for the animals table below, name constraints where applicable at the table level, otherwise name them at the column level. Define the primary key (animal_id). The license_tag_number must be unique. The admit_date and vaccination_date columns cannot contain null values.

animal_id NUMBER(6) **PRIMARY KEY**
name VARCHAR2(25)
license_tag_number NUMBER(10) **UNIQUE**
admit_date DATE **NOT NULL**
adoption_id NUMBER(5),
vaccination_date DATE **NOT NULL**

3. Create the animals table. Write the syntax you will use to create the table.

CREATE TABLE animals (animal_id NUMBER(6) CONSTRAINT pk_animals PRIMARY KEY, name VARCHAR(25), license_tag_number NUMBER(10) CONSTRAINT uq_license UNIQUE, admit_date DATE CONSTRAINT m_admit_date NOT NULL, adoption_id NUMBER(5), vaccination_date DATE CONSTRAINT m_vaccination NOT NULL);

4. Enter one row into the table. Execute a SELECT * statement to verify your input. Refer to the graphic below for input.

ANIMAL_ID	NAME	LICENSE_TAG_NUMBER	ADMIT_DATE	ADOPTION_ID	VACCINATION_DATE
101	Spot	35540	10-Oct-2004	205	12-Oct-2004

INSERT INTO animals VALUES (101, 'spot', 35540, DATE '2004-10-10', 205, DATE '2004-10-12'); SELECT * FROM animals;

5. Write the syntax to create a foreign key (adoption_id) in the animals table that has a corresponding primary-key reference in the adoptions table. Show both the column-level and table-level syntax. Note that because you have not actually created an adoptions table, no adoption_id primary key exists, so the foreign key cannot be added to the animals table.

COLUMN LEVEL: adoption_id NUMBER(5) CONSTRAINT fk_adoption REFERENCES adoptions (adoption_id);

TABLE LEVEL: CONSTRAINT fk_adoption FOREIGN KEY (adoption_id) REFERENCES adoption (adoption_id);

6. What is the effect of setting the foreign key in the ANIMAL table as:

- ON DELETE CASCADE *if an adoption record is deleted, all related animals are also deleted*
- ON DELETE SET NULL *If an adoption ~~set~~ record is deleted, the animal's adoption_id becomes NULL*

7. What are the restrictions on defining a CHECK constraint?

cannot include ^{subqueries} references to columns from other tables

At column level, can only reference that column

At table level, can reference multiple columns, but only within the same row

cannot use non-deterministic functions

NULL VALUES pass the check unless combined with NOT NULL

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	5
Total (15)	15
Faculty Signature	<i>[Signature]</i>