Database Programming with SQL

14-1: Intro to Constraints; NOT NULL and UNIQUE Constraints

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

A **constraint** is a rule enforced on data in a database table to ensure **data integrity** and **consistency**. Constraints prevent invalid data entry and define relationships between tables. Common types of constraints include:

- **NOT NULL:** Ensures a column cannot have null values.
- UNIQUE: Ensures all values in a column are unique.
- **PRIMARY KEY:** Combines NOT NULL and UNIQUE, uniquely identifying each row.
- **FOREIGN KEY:** Enforces referential integrity by ensuring values in a column match values in another table.
- **CHECK:** Ensures values in a column meet a specific condition.
- **DEFAULT:** Assigns a default value to a column if none is provided.
- 2. What are the limitations of constraints that may be applied at the column level and at the table level?

Aspect	Column-Level Constraints	Table-Level Constraints	
Scope	Applied directly to a specific	Can involve multiple columns	
	column.	in the same table.	
Complexity	Limited to single-column	Supports complex conditions	
	constraints like NOT NULL.	like composite keys.	
Flexibility	Easier to use for simple rules.	Required for multi-column	
		constraints like UNIQUE.	
Check Constraint	Applies to a single column.	Can reference multiple	
		columns.	

- 3. Why is it important to give meaningful names to constraints?
 - Clarity: Makes it easier to understand the purpose of a constraint, especially in large databases.
 - **Troubleshooting:** Simplifies debugging and error identification (e.g., CHK PositiveSalary vs. SYS C001234).

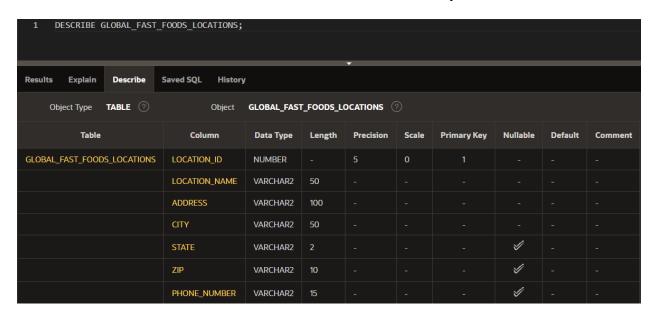
4. Based on the information provided by the owners, choose a datatype for each column. Indicate the length, precision, and scale for each NUMBER datatype.

Column Name	Data Type	Length/ Precision/Scale	Nullable	Reasoning
LOCATION_ID	NUMBER	(5,0)	NOT NULL	Primary Key, unique location identifier.
LOCATION_NAME	VARCHAR2	(50)	NOT NULL	Descriptive name of the location.
ADDRESS	VARCHAR2	(100)	NOT NULL	Location's address.
CITY	VARCHAR2	(50)	NOT NULL	City where the location is situated.
STATE	VARCHAR2	(2)	NULL	State code.
ZIP	VARCHAR2	(10)	NULL	Postal code, may contain alphanumeric characters.
PHONE_NUMBER	VARCHAR2	(15)	NULL	Phone number for the location, can include country/area codes.

- 5. Use "nullable" to indicate those columns that can have null values.
 - Nullable columns: STATE, ZIP, PHONE NUMBER.
 - Non-nullable columns: LOCATION_ID, LOCATION_NAME, ADDRESS, CITY.
- 6. Write the CREATE TABLE statement for the Global Fast Foods locations table to define the constraints at the column level.

```
CREATE TABLE GLOBAL_FAST_FOODS LOCATIONS (
                                     CONSTRAINT PK_LOCATION_ID PRIMARY KEY,
   LOCATION_ID NUMBER(5, 0)
                                    CONSTRAINT NN_LOCATION_NAME NOT NULL,
   LOCATION NAME VARCHAR2(50)
                  VARCHAR2(100)
   ADDRESS
                                    CONSTRAINT NN_ADDRESS NOT NULL,
                  VARCHAR2(50)
   CITY
                                     CONSTRAINT NN CITY NOT NULL,
                  VARCHAR2(2),
                  VARCHAR2(10),
   ZIP
   PHONE NUMBER
                  VARCHAR2(15)
```

- 7. Execute the CREATE TABLE statement in Oracle Application Express.
- 8. Execute a DESCRIBE command to view the Table Summary information.



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.

```
CREATE TABLE GLOBAL FAST FOODS LOCATIONS (
                        NUMBER(5, 0)
         LOCATION ID
                                            CONSTRAINT PK LOCATION ID PRIMARY KEY,
                        VARCHAR2(50)
                                           CONSTRAINT NN LOCATION NAME NOT NULL,
         LOCATION NAME
                        VARCHAR2(100)
                                           CONSTRAINT NN ADDRESS NOT NULL,
         ADDRESS
                        VARCHAR2(50)
                                           CONSTRAINT NN_CITY NOT NULL,
         CITY
                        VARCHAR2(2),
                        VARCHAR2(10),
         ZIP
                        VARCHAR2(15),
         PHONE NUMBER
         CONSTRAINT UQ CITY ZIP UNIQUE (CITY, ZIP)
10
```

14-2: PRIMARY KEY, FOREIGN KEY, and CHECK Constraints

- 1. What is the purpose of:
 - a. PRIMARY KEY: Uniquely identifies each row in a table.
 - b. FOREIGN KEY: Establishes a relationship between two tables by linking a column in one table to the primary key in another table.
 - c. CHECK CONSTRAINT: Ensures that column values meet a specified condition or rule.

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)

name VARCHAR2(25)

license_tag_number NUMBER(10)

admit_date DATE

adoption_id NUMBER(5),

vaccination_date DATE
```

Column Name	Constraint Name	Constraint	Defined
		Type	At
ANIMAL_ID	PK_ANIMAL_ID	PRIMARY	Table
		KEY	Level
LISCENSE_TAG_NUMBER	UQ LISCENSE TAG	UNIQUE	Table
			Level
ADMIT_DATE	NN_ADMIT_DATE	NOT	Column
		NULL	Level
VACCINATION_DATE	NN_VACCINATION_DAT	NOT	Column
_	E	NULL	Level

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

```
CREATE TABLE ANIMALS (
          ANIMAL ID
                                                 CONSTRAINT PK ANIMAL ID PRIMARY KEY,
                               NUMBER(6)
                               VARCHAR2(25),
          LICENSE_TAG_NUMBER NUMBER(10),
          ADMIT DATE
                                                 CONSTRAINT NN ADMIT DATE NOT NULL,
          ADOPTION_ID
                               NUMBER(5),
          VACCINATION DATE
                                                 CONSTRAINT NN VACCINATION DATE NOT NULL,
          CONSTRAINT UQ_LICENSE_TAG_UNIQUE (LICENSE_TAG_NUMBER)
      );
Results
         Explain
                  Describe
                            Saved SQL
                                        History
Table created.
```

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

ANIMAL	NA	LISCENCE_TAG_N	ADMIT_D	ADOPTIO	VACCINATION_
_ID	ME	UMBER	ATE	N_ID	DATE
101	Spot	35540	10-Oct-	205	12-Oct-2004
			2004		

1 SELECT * From Animals								
			*					
Results Exp	Results Explain Describe Saved SQL History							
ANIMAL_ID	NAME	LICENSE_TAG_NUMBER	ADMIT_DATE	ADOPTION_ID	VACCINATION_DATE			
101	Spot	35540	10-Oct-2004	205	12-Oct-2004			

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

```
CREATE TABLE ANIMALS (

ANIMAL_ID NUMBER(6) CONSTRAINT PK_ANIMAL_ID PRIMARY KEY,

NAME VARCHAR2(25),

LICENSE_TAG_NUMBER NUMBER(10),

ADMIT_DATE DATE CONSTRAINT NN_ADMIT_DATE NOT NULL,

ADOPTION_ID NUMBER(5) CONSTRAINT FK_ADOPTION_ID REFERENCES ADOPTIONS (ADOPTION_ID),

VACCINATION_DATE DATE CONSTRAINT NN_VACCINATION_DATE NOT NULL,

CONSTRAINT UQ_LICENSE_TAG_UNIQUE (LICENSE_TAG_NUMBER)

);
```

• Table-Level Syntax:

```
CREATE TABLE ANIMALS (

ANIMAL_ID NUMBER(6) CONSTRAINT PK_ANIMAL_ID PRIMARY KEY,

NAME VARCHAR2(25),

LICENSE_TAG_NUMBER NUMBER(10),

ADMIT_DATE DATE CONSTRAINT NN_ADMIT_DATE NOT NULL,

ADOPTION_ID NUMBER(5),

VACCINATION_DATE DATE CONSTRAINT NN_VACCINATION_DATE NOT NULL,

CONSTRAINT UQ_LICENSE_TAG_UNIQUE (LICENSE_TAG_NUMBER),

CONSTRAINT FK_ADOPTION_ID FOREIGN KEY (ADOPTION_ID) REFERENCES ADOPTIONS (ADOPTION_ID)

10 );
```

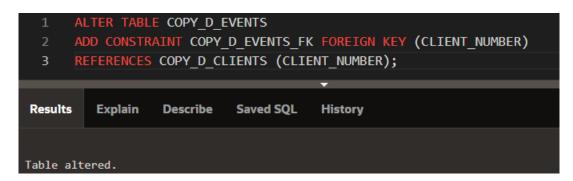
- 6. What is the effect of setting the foreign key in the ANIMAL tables as:
 - a. **ON DELETE CASCADE:** When a referenced record in the parent table is deleted, all rows in the child table that reference the deleted row are automatically deleted.
 - b. **ON DELETE SET NULL:** When a referenced record in the parent table is deleted, all foreign key values in the child table that reference the deleted row are set to NULL.
- 7. What are the restrictions on defining a CHECK constraint?
 - Cannot reference other columns: A CHECK constraint can only validate data within the same row; cross-row or cross-table checks are not supported.
 - Cannot include subqueries: Subqueries or references to other tables are not allowed.
 - Limited functions: Functions like SYSDATE, USER, or PL/SQL constructs cannot be used in a CHECK constraint.

14-3: Managing Constraints

- 1. What are four functions that an ALTER statement can perform on constraints?
 - Add a Constraint
 - **Drop** a Constraint
 - Enable a Constraint
 - **Disable** a Constraint
- 2. Since the tables are copies of the original tables, the integrity rules are not passed onto the new tables; only the column datatype definitions remain. You will need to add a PRIMARY KEY constraint to the copy_d_clients table. Name the primary key copy_d_clients_pk . What is the syntax you used to create the PRIMARY KEY constraint to the copy_d_clients table?



3. Create a FOREIGN KEY constraint in the copy_d_events table. Name the foreign key copy_d_events_fk. This key references the copy_d_clients table client_number column. What is the syntax you used to create the FOREIGN KEY constraint in the copy_d_events table?



- 4. Use a SELECT statement to verify the constraint names for each of the tables. Note that the tablenames must be capitalized.
 - The constraint name for the primary key in the copy_d_clients table is COPY_D_CLIENTS_PK
 - The constraint name for the foreign key in the copy_d_events table is COPY_D_EVENTS_FK
- 5. Drop the PRIMARY KEY constraint on the copy_d_clients table. Explain your results.



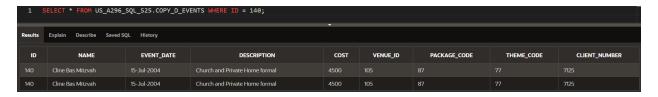
6. Add the following event to the copy d events table. Explain your results.

ID	NAME	EVENT_DA TE	DESCRIPTI ON	COS T	VENUE_I D	PACKAGE_CO DE	THEME_CO DE	CLIENT_NUMB ER
14 0	Cline Bas Mitzva h	15-Jul- 2004	Church and Private Home formal	450 0	105	87	77	7125

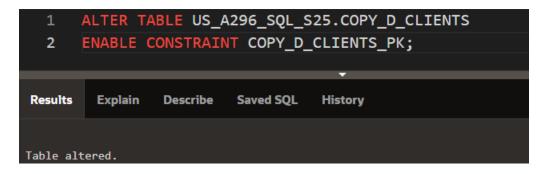


7. Create an ALTER TABLE query to disable the primary key in the copy_d_clients table. Then add the values from #6 to the copy_d_events table. Explain your results.

8. Repeat question 6: Insert the new values in the copy_d_events table. Explain your results.



9. Enable the primary-key constraint in the copy_d_clients table. Explain your results.



- 10. If you wanted to enable the foreign-key column and reestablish the referential integrity between these two tables, what must be done?
 - To reestablish the foreign-key relationship between copy_d_events and copy_d_clients, we need to enable the foreign key constraint in the copy_d_events table.

- 11. Why might you want to disable and then re-enable a constraint?
 - Data Maintenance
 - Performance
 - Fixing Data Integrity Issues
 - Referential Integrity
- 12. Query the data dictionary for some of the constraints that you have created. How does the data dictionary identify each constraint type?
 - Query:

SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE FROM ALL_CONSTRAINTS WHERE OWNER = 'US_A296_SQL_S25' AND TABLE NAME IN ('COPY D CLIENTS', 'COPY D EVENTS');

- Constraint Types:
 - 1. P: Primary Key
 - 2. C: Check Constraint
 - 3. R: Referential Integrity (Foreign Key)