



RELATIONAL DATABASES

CREATE TABLE

SCHEMA OBJECTS

- TABLE
- INDEX
- CONSTRAINT
- VIEW
- SEQUENCE
- SYNONYM

SCHEMA OBJECTS

- SOME OBJECT TYPES TAKE UP SPACE, KNOWN AS STORAGE, OTHERS DO NOT.
- DATABASE OBJECTS TAKING UP SIGNIFICANT STORAGE SPACE ARE KNOWN AS SEGMENTS.
- TABLES AND INDEXES ARE EXAMPLES OF SEGMENTS AS THE VALUES STORED IN THE COLUMNS OF EACH ROW TAKE UP SIGNIFICANT PHYSICAL DISK SPACE.
- VIEWS, CONSTRAINTS, SEQUENCES AND SYNONYMS ARE ALSO OBJECTS BUT THE ONLY SPACE THEY REQUIRE IN THE DATABASE IS IN THE DEFINITION OF THE OBJECT – NONE OF THEM HAVE ANY DATA ROWS ASSOCIATED WITH THEM.

SCHEMA OBJECTS

- THE DATABASE STORES THE DEFINITIONS OF ALL DATABASE OBJECTS IN THE DATA DICTIONARY, AND THESE DEFINITIONS ARE ACCESSIBLE TO ALL USERS OF THE DATABASE AS WELL AS TO THE DATABASE ITSELF.
- THE DATABASE LOOKS UP THE DATA DICTIONARY AND FINDS THE DEFINITION OF THE TABLE USED IN THE QUERY.
- THE DATA DICTIONARY IS USED FOR ALL STATEMENTS ISSUED.
- IT CHECKS THAT THE TABLES YOU ARE REFERENCING IN YOUR STATEMENT EXIST IN THE DATABASE, IT CHECKS THE COLUMN NAMES ARE CORRECT, AND THAT YOU HAVE THE CORRECT PRIVILEGES TO PERFORM THE ACTION YOU ARE REQUESTING AND FINALLY IT USES THE DICTIONARY TO DECIDE THE EXECUTION PLAN – HOW IT WILL PERFORM THE REQUEST.

SCHEMA OBJECTS

- THE DATA DICTIONARY CAN BE QUERIED BY ALL DATABASE USERS.
- IN MOST DATABASE APPLICATIONS, IT CAN BE ACCESSED BOTH VIA SQL STATEMENTS AND ALSO WITHIN THE GUI INTERFACE THROUGH A BROWSER OF OBJECTS.
- USING A SQL STATEMENT, YOU HAVE TO KNOW THE NAMES OF THE TABLE YOU ARE QUERYING.
- IN THE INTERFACE YOU SIMPLY CLICK THE LISTED OBJECTS TO SEE THEIR DETAILS.

DATA DICTIONARY

- TWO KINDS OF TABLES EXIST IN THE DATA DICTIONARY, USER TABLES AND DATA DICTIONARY TABLES.
- YOU CAN USE SQL STATEMENTS TO ACCESS BOTH KINDS OF TABLES – YOU CAN SELECT, INSERT, UPDATE, AND DELETE DATA IN THE USER TABLES, AND YOU CAN SELECT DATA IN THE DATA DICTIONARY TABLES.
- USER TABLES: EMPLOYEES, DEPARTMENTS, JOBS ETC
- DATA DICTIONARY TABLES: INFORMATION_SCHEMA.TABLES, ETC
- DATA DICTIONARY TABLES ARE ALL OWNED BY ROOT AND ONLY SELECT STATEMENTS SHOULD BE USED WITH ANY OF THESE TABLES.

DATA DICTIONARY

- WHEN YOU ARE USING THE DATA DICTIONARY YOU MUST KNOW THE NAMES OF WHAT YOU WISH TO SEE.
- IN MYSQL WORKSHOP THIS IS QUITE SIMPLE:

```
SELECT * FROM INFORMATION_SCHEMA.TABLES;
```

```
SELECT * FROM INFORMATION_SCHEMA.COLUMNS;
```

TABLE CREATION

- ALL DATA IN A RELATIONAL DATABASE IS STORED IN TABLES.
- WHEN CREATING A NEW TABLE, UNQUOTED NAMES :
 - 1 TO 64 CHARACTERS LONG
 - MUST CONTAIN ALPHANUMERIC
 - CAN CONTAIN _ AND \$
 - CANNOT CONTAIN .
- QUOTED NAMES:
 - CAN CONTAIN ANY CHARACTERS
- MUST NOT DUPLICATE THE NAME OF ANOTHER OBJECT OWNED BY THE SAME USER
- MUST NOT BE A MQL RESERVED WORD.
- TABLE NAMES ARE ONLY CASE SENSITIVE IF THE OPERATING SYSTEM OF THE DATABASE SERVER IS UNIX,
- THEY SHOULD BE PLURAL

DATA DEFINITION LANGUAGE (DDL)

- CREATING TABLES IS PART OF SQL'S DATA DEFINITION LANGUAGE (DDL).
- OTHER DDL STATEMENTS USED TO SET UP, CHANGE, AND REMOVE DATA STRUCTURES FROM TABLES INCLUDE ALTER, DROP, RENAME, AND TRUNCATE.
- TO CREATE A NEW TABLE, YOU MUST HAVE THE CREATE TABLE PRIVILEGE AND A STORAGE AREA FOR IT.
- THE DATABASE ADMINISTRATOR USES DATA CONTROL LANGUAGE (DCL) STATEMENTS TO GRANT THIS PRIVILEGE TO USERS AND ASSIGN A STORAGE AREA.
- TABLES BELONGING TO OTHER USERS ARE NOT IN YOUR SCHEMA (STORAGE AREA).
- IF YOU WANT TO VIEW A TABLE THAT IS NOT IN YOUR SCHEMA YOU USE THE TABLE OWNERS NAME AS A PREFIX TO THE TABLE NAME.

CREAT TABLE SYNTAX

- TO CREATE A NEW TABLE, USE THE FOLLOWING SYNTAX DETAILS:
 - TABLE IS THE NAME OF THE TABLE.
 - COLUMN IS THE NAME OF THE COLUMN
 - DATA TYPE IS THE COLUMN'S DATA TYPE AND LENGTH
 - DEFAULT EXPRESSION SPECIFIES A DEFAULT VALUE IF A VALUE IS OMITTED WHEN DATA IS INSERTED.

```
CREATE TABLE table  
(column data type [DEFAULT expression],  
 (column data type [DEFAULT expression],  
 (.....[ ] );
```

CREATE TABLE EXAMPLE

```
CREATE TABLE my_cd_collection  
(cd_number INT(3),  
title VARCHAR(20),  
artist VARCHAR(20),  
purchase_date DATE DEFAULT CURRENT_DATE());
```

```
CREATE TABLE my_friends  
(first_name VARCHAR(20),  
Last_name VARCHAR(20),  
email VARCHAR(30),  
Phone_num VARCHAR(12),  
birth_date DATE );
```