United States International University-Africa

DATABASE SYSTEMS

Presented by Prof. Elisha T. O. Omulo

USIU AFRICA

WEEK 5 AGENDA

Sub topics(Activities/ Session tasks/Learning Sub Outcomes):

- **The history and role of SQL in database development.**
- **Creating a database using the SQL data definition language.**
- **Single-table queries using SQL commands.**
- **Enforcing referential integrity using SQL.**
- **The SQL: 1999 and SQL: 200n standards.**

Course Textbook: **Carlos Coronel, Steven Morris, Peter Rob and Keeley Crockett Database Principles: Fundamentals of Design, Implementation, and Management, 14th Edition, 2022, ISBN-13978-0357673034.**

The history and role of SQL in database development.

Introductory remarks

Database language expectations- enable:

- **Creation of the database and relation structures;**
- **Performing basic data management tasks, such as the insertion, modification, and deletion of data from the relations;**
- **Perform both simple and complex queries.**
- **Users to apply effort, and its command structure and syntax must be relatively easy to learn.**
- **Conformity with some recognized standard so that we can use the same command structure and syntax when we move from one DBMS to another.**

SQL is intended to satisfy these requirements.

Task: Describe what a database language is expected to do.

The history and role of SQL in database development.

Introductory remarks

SQL is designed to use relations to transform inputs into required outputs. As a language, the ISO SQL standard has two major components:

- **a Data Definition Language (DDL) for defining (creating or setting up) the database structure and controlling access to the data;**

- **a Data Manipulation Language (DML) for retrieving and updating data.**

Task: discuss the components of a database database language.

The history and role of SQL in database development.

Introductory remarks

Data Manipulation- Until SQL:1999, SQL contained only these definitional and manipulative commands; after this in now had flow of control commands, such as IF . . . THEN . . . ELSE, GO TO, or DO . . . WHILE. These had to be implemented using a programming or job-control language, or interactively by the decisions of the user.

So SQL can be used in two ways.

- **The first way is to use SQL interactively by entering the statements at a terminal.**
- **The second way is to embed SQL statements in a procedural language.**

Task: discuss "from definitional to flow of controls" Expectations?

The history and role of SQL in database development.

Introductory remarks

SQL is a relatively easy language to learn:

- **It is a non-procedural language: you specify what information you require, rather than how to get it. In other words, SQL does not require you to specify the access methods to the data.**

- **As a modern languages, SQL is essentially free-format, which means that parts of statements do not have to be typed at particular locations on the screen.**

- **The command structure consists of standard English words such as CREATE TABLE, INSERT, SELECT.**

Task: Is SQL an easy language to learn?

The history and role of SQL in database development.

Introductory remarks

SQL is a relatively easy language to learn:

Example:

– CREATE TABLE Staff (staffNo VARCHAR(5), lName VARCHAR(15), salary DECIMAL(7,2));
– INSERT INTO Staff VALUES ('SG16', 'Guraj', 8300);
– SELECT staffNo, lName, salary
FROM Staff
WHERE salary > 10000;

- **SQL users: many- Database Administrators (DBA), management personnel, application developers and many other types of end-user.**

- **An international standard exists for the SQL; it is both the formal and de facto standard language for defining and manipulating relational databases (ISO, 1992, 1999a).**

Task: Formulate SQL commands to create a STUDENT table and enter 3 records

The history and role of SQL in database development.

The History

Origin: 1970-the publication of the seminal paper by E. F. Codd, while working at IBM's Research Laboratory in San José (Codd, 1970)

- **1974: D. Chamberlin, from the IBM San José Laboratory, defined a language called the Structured English Query Language, or SEQUEL.**
- **1976: A revised version, SEQUEL/2, was defined, renamed SQL, pronounce as 'S-Q-L'. Also IBM produced a prototype DBMS based on SEQUEL/2, called System R that was a very successful proof of concept.**
- **However, the roots of SQL are in the language SQUARE (Specifying Queries As Relational Expressions), which pre-dates the System R project.**
- **SQUARE was designed as a research language to implement relational algebra with English sentences.**

Task: discus the origin of SQL until the late 1970s.

The history and role of SQL in database development.

The History

1970s:

- **Late, Oracle DBMS was produced by now called the Oracle Corporation, being the first commercial implementation of a relational DBMS based on SQL.**
- **INGRES followed shortly afterwards, with a query language called QUEL, which was more 'structured' than SQL, but was less English-like. Later INGRES was converted to an SQL-based DBMS.**

1981/1982:

- **IBM produced its first commercial RDBMS, called SQL/ DS, for the DOS/VSE and VM/CMS environments.**

1983: IBM developed as DB2 for the MVS.

Task: discus the origin of SQL in the 1980s.

The history and role of SQL in database development.

The History

Standards:

1982: American National Standards Institute started work on a Relational Data-base Language (RDL) based on a concept paper from IBM.

1983: ISO joined standards work and together they defined a standard for SQL.

1984: The name RDL was dropped, and the draft standard reverted to a form that was more like the existing implementations of SQL.

1987: The initial ISO standard published, and it attracted a considerable degree of criticism from C. Date, that it lacked some essential features.

1989: ISO published an addendum that defined an 'Integrity Enhancement Feature'.

Task: discus the evolution of the SQL standards.

The history and role of SQL in database development- The History

Standards:

1992:First major revision to the ISO standard, the SQL2 or SQL-92

1999:new release of the standard, SQL:1999, it had additional features to support object-oriented data management.

2003: a further release, SQL:2003 was available. It had vendors extensions leading to dialects. SQL:2003 has Core SQL features that a vendor must implement to claim comliance with the SQL:2003 standard. Other many remaining features were divided into packages eg. packages for object features and OLAP (OnLine Analytical Processing).

2016: SQL:2016 or ISO/IEC 9075:2016 – allows JSON document access; Row Pattern Recognition; Date and time formatting and parsing; LISTAGG- group of rows convertible to a delimited string; Polymorphic table functions with no predefined return type; new data type DECFLOAT. It had XML-Related Specifications (SQL/XML).

2019: ISO/IEC 9075-15:2019 Part 15: Multi-dimensional arrays (SQL/MDA).

Today: SQL was originally an IBM concept, but now other vendors joined and there are literally hundreds of SQL-based products available.

Task: discus the current SQL standards.

Creating a database using the SQL data definition language.

General SQL commands: **A database language allows you to create database and table structures, perform basic data management chores (add, delete, and modify), and perform complex queries designed to transform the raw data into information. It must be easy to use.**

COMMAND OR OPTION- DATA DEFINITION(CREATION) COMMANDS

CREATE SCHEMA- Creates a database schema

AUTHORIZATION

CREATE TABLE-Creates a new table in the user's database schema

NOT NULL- Ensures that a column will not have null values

UNIQUE- Ensures that a column will not have duplicate values

PRIMARY KEY-Defines a primary key for a table

FOREIGN KEY-Defines a foreign key for a table

DEFAULT- Defines a default value for a column (when no value is given)

CHECK-Validates data in an attribute

Task: **Discuss where you would use the given commands.**

Creating a database using the SQL data definition language.

General SQL commands:

COMMAND OR OPTION- DATA DEFINITION(CREATION) COMMANDS

- **CREATE INDEX-Creates an index for a table**
- **CREATE VIEW-Creates a dynamic subset of rows and columns from one or more tables**
- **ALTER TABLE- Modifies a table's definition (adds, modifies, or deletes attributes or constraints)**
- **CREATE TABLE AS Creates a new table based on a query in the user's database schema**
- **DROP TABLE- Permanently deletes a table (and its data)**
- **DROP INDEX- Permanently deletes an index**
- **DROP VIEW- Permanently deletes a view**

Task: **Discuss how you would use these commands**

Creating a database using the SQL data definition language.

General SQL commands:

COMMAND OR OPTION- DATA MANIPULATION(PROCESSING) COMMANDS

- **INSERT-Inserts row(s) into a table**
- **SELECT- Selects attributes from rows in one or more tables or views**
    - **WHERE - Restricts the selection of rows based on a conditional expression**
    - **GROUP BY- Groups the selected rows based on one or more attributes**
    - **HAVING -Restricts the selection of grouped rows based on a condition**
    - **ORDER BY- Orders the selected rows based on one or more attributes**
- **UPDATE- Modifies an attribute's values in one or more table's rows**
- **DELETE-Deletes one or more rows from a table**
- **COMMIT- Permanently saves data changes**
- **ROLLBACK- Restores data to their original values**

Comparison operators

- **=, <, >, <=, >=, <> Used in conditional expressions**

Task: What would one do with data processing commands? Discuss.

Creating a database using the SQL data definition language.

General SQL commands:

COMMAND OR OPTION- DATA MANIPULATION(PROCESSING) COMMANDS

Logical operators: AND/OR/NOT- Used in conditional expressions

Special operators

- **BETWEEN - Checks whether an attribute value is within a range**
- **IS NULL- Checks whether an attribute value is null**
- **LIKE- Checks whether an attribute value matches a given string pattern**
- **IN- Checks whether an attribute value matches any value within a value list**
- **EXISTS- Checks whether a subquery returns any rows**
- **DISTINCT- Limits values to unique values**

Aggregate functions- Used with SELECT to return mathematical summaries on columns

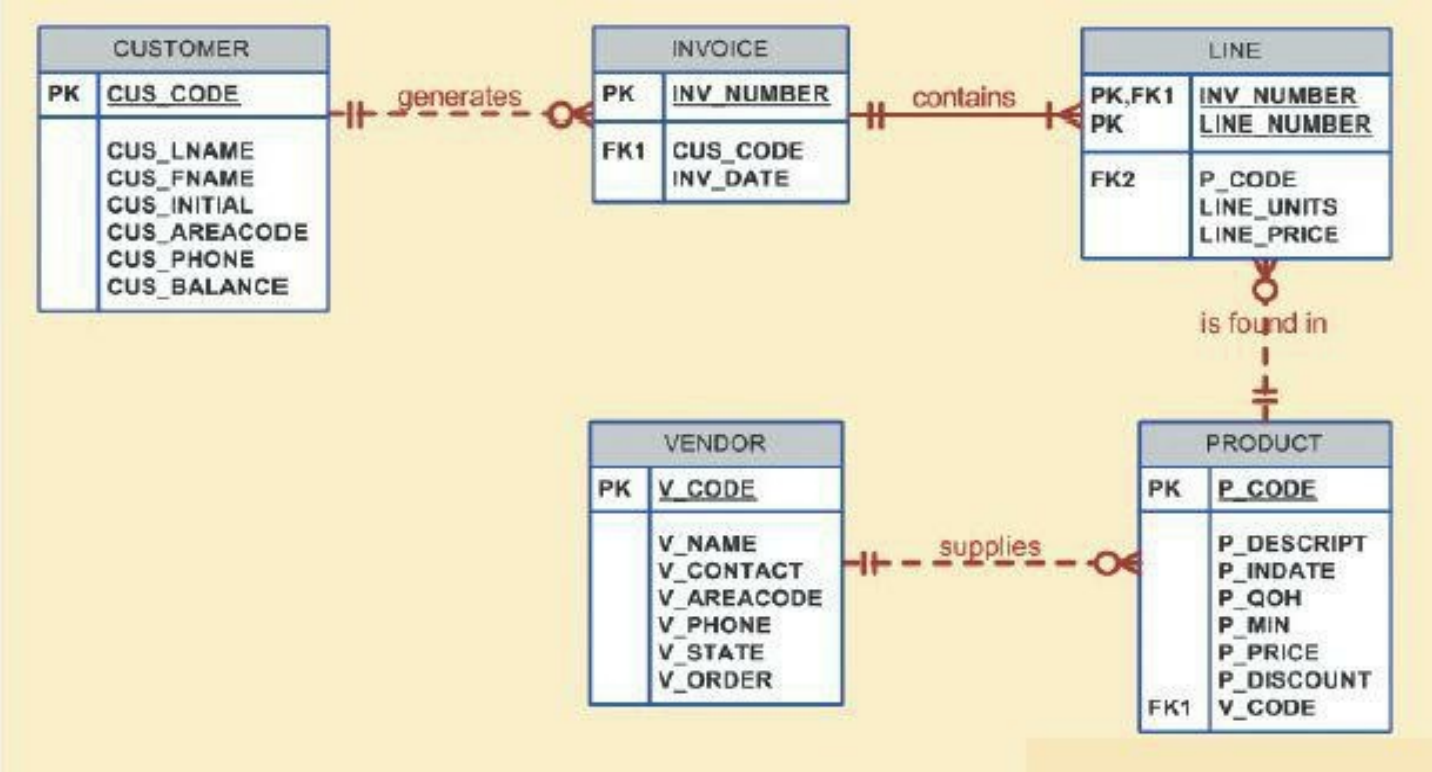
- **COUNT-Returns the number of rows with non-null values for a given column**
- **MIN- Returns the minimum attribute value found in a given column**
- **MAX-Returns the maximum attribute value found in a given column**
- **SUM-Returns the sum of all values for a given column**
- **AVG-Returns the average of all values for a given column**

Task: What would one do with data processing commands? Discuss.

Creating a database using the SQL data definition language.

The Database Model to be used:

A simple database composed of the following tables is used to illustrate the SQL commands in this chapter: CUSTOMER, INVOICE, LINE, PRODUCT, and VENDOR.



- A customer may generate many invoices. Each invoice is generated by one customer.
- An invoice contains one or more invoice lines. Each invoice line is associated with one invoice.
- Each invoice line references one product. A product may be found in many invoice lines. (You can sell more than one hammer to more than one customer.)
- A vendor may supply many products. Some vendors do not yet supply products. For example, a vendor list may include potential vendors.
- If a product is vendor supplied, it is supplied by only a single vendor.
- Some products are not supplied by a vendor.

Task: **Discuss the database schema that is provided.**

Creating a database using the SQL data definition language.

The Database Tables

Table name: VENDOR **Database name: Ch07_SaleCo**

V_CODE	V_NAME	V_CONTACT	V_AREACODE	V_PHONE	V_STATE	V_ORDER
21225	Bryson, Inc.	Smithson	615	223-3234	TN	Y
21226	SuperLoo, Inc.	Flushing	904	215-8995	FL	N
21231	D&E Supply	Singh	615	228-3245	TN	Y
21344	Gomez Bros.	Ortega	615	889-2546	KY	N
22567	Dome Supply	Smith	901	678-1419	GA	N
23119	Randsets Ltd.	Anderson	901	678-3998	GA	Y
24004	Brackman Bros.	Browning	615	228-1410	TN	N
24288	ORDVA, Inc.	Hakford	615	898-1234	TN	Y
25443	B&K, Inc.	Smith	904	227-0093	FL	N
25501	Damal Supplies	Smythe	615	890-3529	TN	N
25595	Rubicon Systems	Orton	904	456-0092	FL	Y

Table name: PRODUCT

P_CODE	P_DESCRIPT	P_INDATE	P_QOH	P_MIN	P_PRICE	P_DISCOUNT	V_CODE
11QER/31	Power painter, 15 psi., 3-nozzle	03-Nov-11	8	5	109.99	0.00	25595
13-Q2/P2	7.25-in. pwr. saw blade	13-Dec-11	32	15	14.99	0.05	21344
14-Q1/L3	9.00-in. pwr. saw blade	13-Nov-11	18	12	17.49	0.00	21344
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-Jan-12	15	8	39.95	0.00	23119
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-Jan-12	23	5	43.99	0.00	23119
2232/QTY	B&D jigsaw, 12-in. blade	30-Dec-11	8	5	109.92	0.05	24288
2232/QWE	B&D jigsaw, 8-in. blade	24-Dec-11	6	5	99.87	0.05	24288
2238/QPD	B&D cordless drill, 1/2-in.	20-Jan-12	12	5	38.95	0.05	25595
23109-HB	Claw hammer	20-Jan-12	23	10	9.95	0.10	21225
23114-AA	Sledge hammer, 12 lb.	02-Jan-12	8	5	14.40	0.05	
54778-2T	Rat-tail file, 1/8-in. fine	15-Dec-11	43	20	4.99	0.00	21344
89-WRE-Q	Hicut chain saw, 16 in.	07-Feb-12	11	5	256.99	0.05	24288
PVC23DRT	PVC pipe, 3.5-in., 8-ft	20-Feb-12	188	75	5.87	0.00	
SM-18277	1.25-in. metal screw, 25	01-Mar-12	172	75	6.99	0.00	21225
SW-23116	2.5-in. wd. screw, 50	24-Feb-12	237	100	8.45	0.00	21231
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-Jan-12	18	5	119.95	0.10	25595

- **The VENDOR table contains vendors who are not referenced in the PRODUCT table.**
- **Database designers note that possibility by saying that PRODUCT is optional to VENDOR; a vendor may exist without a reference to a product.**
- **Existing V_CODE values in the PRODUCT table must (and do) have a match in the VENDOR table to ensure referential integrity.**
- **A few products are supplied factory-direct, a few are made in-house, and a few may have**
- **been bought in a warehouse sale. In other words, a product is not necessarily supplied by a vendor. Therefore, VENDOR is optional to PRODUCT.**

Task: Discuss the database tables that are provided.

Creating a database using the SQL data definition language.

Creating(defining) the Database

To use a new RDBMS, you must do 2 things:

- **create the database structure (schema)**
- **create the tables that will hold the end-user data.**

- **For the database structure: the RDBMS creates the physical files that will hold the database;**
- **RDBMS automatically creates the data dictionary tables in which to store the metadata and creates a default database administrator.**

- **Creating the database structure differs substantially from one RDBMS to another; but , it is relatively easy to create a database structure, regardless of which RDBMS you use.**

Task: Discuss the database tables that are provided.

Creating a database using the SQL data definition language.

Creating(defining) the Database

- **Microsoft Access: start Access, click the File tab, click want to store the database, and then name the database.**
- **Oracle, MS SQL Server, MySQL, or DB2: Due to security requirements and greater complexity, creating a database is a more elaborate process.**
- **Some RDBMS vendors use SQL that deviates little from the ANSI standard SQL eg. most RDBMSs require each SQL command to end with a semicolon, but, some SQL implementations do not use a semicolon.**
- **For enterprise RDBMS, you must be authenticated by the RDBMS before you can start creating tables.**
- **Authentication is the process the DBMS uses to verify that only registered users access the database.**
- **To be authenticated, you must log on to the RDBMS using a user ID and a password created by the database administrator. In an enterprise RDBMS, every user ID is associated with a database schema.**

USIU AFRICA

Task: Discuss the database tables that are provided.

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

- **CREATE SCHEMA AUTHORIZATION {creator}; if the creator is LYDIA, the following command is used: CREATE SCHEMA AUTHORIZATION LYDIA;**

- **Most enterprise RDBMSs support that command, but the command is rarely used directly, from the command line.**
- **When a user is created, the DBMS automatically assigns a schema to that user.**
- **When the DBMS is used, the CREATE SCHEMA AUTHORIZATION command must be issued by the user who owns the schema.**
- **That is, if you log on as LYDIA, you can only use CREATE SCHEMA AUTHORIZATION LYDIA.**
- **For most RDBMSs, the CREATE SCHEMA AUTHORIZATION command is optional, and we focus on the ANSI SQL commands required to create and manipulate tables.**

Task: Discuss the database structure creation in a selected DBMS.

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

- **Remember, the data types selected depend on the nature of the data and type of use.**

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE*	REQUIRED	PK OR FK	FK REFERENCED TABLE
PRODUCT	P_CODE	Product code	VARCHAR(10)	XXXXXXXXXX	NA	Y	PK	
	P_DESCRIPT	Product description	VARCHAR(35)	Xxxxxxxxxxxx	NA	Y		
	P_INDATE	Stocking date	DATE	DD-MON-YYYY	NA	Y		
	P_QOH	Units available	SMALLINT	####	0–9999	Y		
	P_MIN	Minimum units	SMALLINT	####	0–9999	Y		
	P_PRICE	Product price	NUMBER(8,2)	####.##	0.00–9999.00	Y		
	P_DISCOUNT	Discount rate	NUMBER(5,2)	0.##	0.00–0.20	Y		
	V_CODE	Vendor code	INTEGER	###	100–999		FK	VENDOR
VENDOR	V_CODE	Vendor code	INTEGER	#####	1000–9999	Y	PK	
	V_NAME	Vendor name	VARCHAR(35)	Xxxxxxxxxxxxxx	NA	Y		
	V_CONTACT	Contact person	VARCHAR(25)	Xxxxxxxxxxxxxx	NA	Y		
	V_AREACODE	Area code	CHAR(3)	999	NA	Y		
	V_PHONE	Phone number	CHAR(8)	999–9999	NA	Y		
	V_STATE	State	CHAR(2)	XX	NA	Y		
	V_ORDER	Previous order	CHAR(1)	X	Y or N	Y		

Task: **Discuss the details around data dictionary.**

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

- **Remember, the data types selected depend on the nature of the data and type of use.**

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE*	REQUIRED	PK OR FK	FK REFERENCED TABLE
PRODUCT	P_CODE	Product code	VARCHAR(10)	XXXXXXXXX	NA	Y	PK	
	P_DESCRIPT	Product description	VARCHAR(35)	Xxxxxxxxxxx	NA	Y		
	P_INDATE	Stocking date	DATE	DD-MON-YYYY	NA	Y		
	P_QOH	Units available	SMALLINT	####	0–9999	Y		
	P_MIN	Minimum units	SMALLINT	####	0–9999	Y		
	P_PRICE	Product price	NUMBER(8,2)	####.##	0.00–9999.00	Y		
	P_DISCOUNT	Discount rate	NUMBER(5,2)	0.##	0.00–0.20	Y		
	V_CODE	Vendor code	INTEGER	###	100–999		FK	VENDOR
VENDOR	V_CODE	Vendor code	INTEGER	#####	1000–9999	Y	PK	
	V_NAME	Vendor name	VARCHAR(35)	Xxxxxxxxxxxxx	NA	Y		
	V_CONTACT	Contact person	VARCHAR(25)	Xxxxxxxxxxxxx	NA	Y		
	V_AREACODE	Area code	CHAR(3)	999	NA	Y		
	V_PHONE	Phone number	CHAR(8)	999–9999	NA	Y		
	V_STATE	State	CHAR(2)	XX	NA	Y		
	V_ORDER	Previous order	CHAR(1)	X	Y or N	Y		

- **FK= Foreign key; PK= Primary key**
- **CHAR= Fixed-length character data, 1 to 255 characters;**
- **VARCHAR= Variable-length character data, 1 to 2,000 characters. VARCHAR is automatically converted to VARCHAR2 in Oracle.**
- **NUMBER= Numeric data. NUMBER(9,2) is used to specify numbers that have two decimal places and are up to nine digits long, including the decimal places.**
- **Some RDBMSs permit the use of a MONEY or a CURRENCY data type.**

Task: **Discuss the details around data dictionary.**

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

- **Remember, the data types selected depend on the nature of the data and type of use.**

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE*	REQUIRED	PK OR FK	FK REFERENCED TABLE
PRODUCT	P_CODE	Product code	VARCHAR(10)	XXXXXXXXX	NA	Y	PK	
	P_DESCRIPT	Product description	VARCHAR(35)	Xxxxxxxxxxx	NA	Y		
	P_INDATE	Stocking date	DATE	DD-MON-YYYY	NA	Y		
	P_QOH	Units available	SMALLINT	####	0–9999	Y		
	P_MIN	Minimum units	SMALLINT	####	0–9999	Y		
	P_PRICE	Product price	NUMBER(8,2)	####.##	0.00–9999.00	Y		
	P_DISCOUNT	Discount rate	NUMBER(5,2)	0.##	0.00–0.20	Y		
	V_CODE	Vendor code	INTEGER	###	100–999		FK	VENDOR
VENDOR	V_CODE	Vendor code	INTEGER	#####	1000–9999	Y	PK	
	V_NAME	Vendor name	VARCHAR(35)	Xxxxxxxxxxxxx	NA	Y		
	V_CONTACT	Contact person	VARCHAR(25)	Xxxxxxxxxxxxx	NA	Y		
	V_AREACODE	Area code	CHAR(3)	999	NA	Y		
	V_PHONE	Phone number	CHAR(8)	999–9999	NA	Y		
	V_STATE	State	CHAR(2)	XX	NA	Y		
	V_ORDER	Previous order	CHAR(1)	X	Y or N	Y		

- **INT= Integer values only. INT is automatically converted to NUMBER in Oracle.**
- **SMALLINT= Small integer values only. SMALLINT is automatically converted to NUMBER in Oracle.**
- **DATE formats vary. Commonly accepted formats are DD-MON-YYYY, DD-MON-YY, MM/DD/YYYY, and MM/DD/YY.**

Task: **Discuss the details around data dictionary.**

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

Main data types-
Character

- **CHAR(L): Fixed-length character data for up to 255 characters. If you store strings that are not as long as the CHAR parameter value, the remaining spaces are left unused.**

- **VARCHAR(L): if you specify CHAR(25), strings such as Smith and Katzenjammer are each stored as 25 characters. However, some code may always be three digits long, so CHAR(3) would be appropriate if you wanted to store such codes.**

- **VARCHAR2(L): Variable-length character data. The designation VARCHAR2(25) will let you Store characters up to 25 characters long. However, VARCHAR will not leave unused spaces. Oracle automatically converts VARCHAR to VARCHAR2.**

Task: Discuss the details around data dictionary.

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

Main data types-

NUMBER(L,D) with two decimal places and may be up to seven digits long, including the sign and the decimal place (for example, 12.32 or -134.99).

INTEGER: abbreviated as INT. Integers are (whole) counting numbers, so they cannot be used if you want to store numbers that require decimal places.

SMALLINT: Like INTEGER but limited to integer values up to six digits. If your integer values are relatively small, use SMALLINT instead of INT.

DECIMAL(L,D): Like the NUMBER specification, but the storage length is a minimum Specification ie. to enter greater lengths are acceptable, but not enter smaller ones. DECIMAL(9,2), DECIMAL(9), and DECIMAL are all acceptable.

Task: Discuss the details around data dictionary.

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

General Syntax

We need to create the PRODUCT and VENDOR table structures with the help of SQL, we use the CREATE TABLE syntax:

```
CREATE TABLE tablename (
        column1      data type[constraint] [,
        column2      data type[constraint]] [,
PRIMARY KEY (column1 [, column2])] [,
FOREIGN KEY (column1 [, column2]) REFERENCES tablename] [,
CONSTRAINT constraint]);
```

Task: Discuss why we need the create table command.

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

Implementation of vendor table

```
CREATE TABLE VENDOR (
        V_CODE          INTEGER         NOT NULL UNIQUE,
        V_NAME          VARCHAR(35)     NOT NULL,
        V_CONTACT       VARCHAR(25)     NOT NULL,
        V_AREACODE      CHAR(3)         NOT NULL,
        V_PHONE         CHAR(8)         NOT NULL,
        V_STATE         CHAR(2)         NOT NULL,
        PRIMARY KEY (V_CODE));
```

Task: Discuss how to the vendor table; create two more tables.

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

Implement product table-

```
CREATE TABLE PRODUCT (
        P_CODE                  VARCHAR(10) NOT NULL UNIQUE,
        P_DESCRIPT              VARCHAR(35) NOT NULL,
        P_INDATE                DATE NOT NULL,
        P_QOH                   SMALL INT NOT NULL,
        P_MIN                   SMALL INT NOT NULL,
        P_PRICE                 NUMBER(8,2) NOT NULL,
        P_DISCOUNT              NUMBER(8,2) NOT NULL,
        V_CODE                  INTEGER,
PRIMARY KEY (P_CODE),
FOREIGN KEY (V_CODE) REFERENCES VENDOR ON
UPDATE CASCADE);
```

Task: create three different tables.

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

Implement product table-ORACLE-When you press Enter after typing each line, a line number is automatically generated as long as you do not type a semicolon before pressing Enter. The CREATE TABLE command will look like this:

CREATE TABLE PRODUCT (

```
2   P_CODE       VARCHAR2(10)
3   CONSTRAINT   PRODUCT_P_CODE_PK PRIMARY KEY,
4   P_DESCRIPT   VARCHAR2(35)          NOT NULL,
5   P_INDATE     DATE                  NOT NULL,
6   P_QOH        NUMBER                NOT NULL,
7   P_MIN        NUMBER                NOT NULL,
8   P_PRICE      NUMBER(8,2)           NOT NULL,
9   P_DISCOUNT   NUMBER(5,2)           NOT NULL,
10  V_CODE       NUMBER,
11  CONSTRAINT   PRODUCT_V_CODE_FK
12  FOREIGN KEY  V_CODE REFERENCES VENDOR)
13  ;
```

Task: create all other tables in our schema.

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

Implement product table-ORACLE-When you press Enter after typing each line, a line number is automatically generated as long as you do not type a semicolon before pressing Enter. The CREATE TABLE command will look like this:

CREATE TABLE CUSTOMER (

```
CREATE TABLE CUSTOMER (
CUS_CODE          NUMBER          PRIMARY KEY,
CUS_LNAME         VARCHAR(15)     NOT NULL,
CUS_FNAME         VARCHAR(15)     NOT NULL,
CUS_INITIAL       CHAR(1),
CUS_AREACODE      CHAR(3)         DEFAULT '615'     NOT NULL
                                  CHECK(CUS_AREACODE IN ('615','713','931')),
CUS_PHONE         CHAR(8)         NOT NULL,
CUS_BALANCE       NUMBER(9,2)     DEFAULT 0.00,
CONSTRAINT CUS_UI1 UNIQUE (CUS_LNAME, CUS_FNAME));
```

Task: create all other tables in our schema.

Creating a database using the SQL data definition language.

Creating(defining) the Database- ANSI SQL standards define a command to create a database schema:

MS Access- MS Access does not accept the DEFAULT or CHECK constraints. However, MS Access will accept the CONSTRAINT CUS_UI1 UNIQUE (CUS_LNAME, CUS_FNAME) line and create the unique index.

Task: create all other tables in our schema.

Single-table queries using SQL commands.

DATA MANIPULATION COMMANDS

These include:

INSERT – to insert data into a table;

SELECT – to query data in the database;

UPDATE – to update data in a table;

DELETE – to delete data from a table.

INSERT

The INSERT command to enter data into a table.

The INSERT command's basic syntax looks like this:

INSERT INTO tablename VALUES (value1, value2, ..., valuen)

Example:

```
INSERT INTO VENDOR
        VALUES (21225,'Bassey, Inc.','Nairobi','615','223-3234','Tt','Y');
INSERT INTO VENDOR
        VALUES (21226,'Mwangi, Inc.','Valley','904','215-8995','Ft','N');
```

Task: **Insert at least 5 values in to the vendor table.**

Single-table queries using SQL commands.

DATA MANIPULATION COMMANDS

These include: **INSERT – to insert data into a table; SELECT – to query data in the database; UPDATE – to update data in a table DELETE – to delete data from a table.**

INSERT

Example(NULL):

```
INSERT INTO PRODUCT
        VALUES ('BRT-345','Titanium drill bit','18-Oct-11', 75, 10, 4.50, 0.06, NULL);
```

- **ONLY possible where the attribute did not have not null specificition.**
- **The insert command adds only one row or record to the table.**

Inserting Rows with Optional Attributes

If more than one attribute is optional, then you can just enter or use INSERT command, for the attributes that have required values.eg. attributes for the PRODUCT table are P_CODE and P_DESCRIPT:

```
INSERT INTO PRODUCT(P_CODE, P_DESCRIPT) VALUES ('BRT-345','Titanium drill bit');
```

Task: **Insert at least 5 values in to the vendor table.**

Single-table queries using SQL commands.

DATA MANIPULATION COMMANDS

These include: **INSERT – to insert data into a table; SELECT – to query data in the database; UPDATE – to update data in a table DELETE – to delete data from a table.**

SAVING TABLE CHANGES

- **Changes made to the table contents are not saved on disk until you close the database close the program you are using, or use the COMMIT command.**
- **If the database is open and a power outage or some other interruption occurs before you issue the COMMIT command, your changes will be lost and only the original table contents will be retained.**

The syntax for the COMMIT command is:

COMMIT [WORK]

- **The COMMIT command permanently saves all changes—such as rows added, attributes modified, and rows deleted—made to any table in the database.**
- **Therefore, if you intend to make your changes to the PRODUCT table permanent, it is a good idea to save those changes by using the following command:**

COMMIT;

Task: Discuss the need for using the command COMMIT.

Single-table queries using SQL commands.

DATA MANIPULATION COMMANDS

These include: **INSERT – to insert data into a table; SELECT – to query data in the database; UPDATE – to update data in a table DELETE – to delete data from a table.**

THE SELECT COMMAND

- **The SELECT command is used to list the contents of a table. The syntax of the SELECT command is as follows:**

  SELECT columnlist FROM tablename

- **The columnlist represents one or more attributes, separated by commas.**
- **The asterisk (*) as a wildcard character to list all attributes. Eg. SELECT * FROM PRODUCT;**

In ORACLE: Some SQL implementations (such as Oracle's) cut the attribute labels to fit the width of the column.

- **However, Oracle lets you set the width of the display column to show the complete attribute name. You can also change the display format, regardless of how the data are stored in the table, eg. EG to display dollar symbols and commas in the P_PRICE Output you can declare: COLUMN P_PRICE FORMAT $99,999.99 chnging 12347.67 to $12,347.67.**
- **In the same manner, to display only the first 12 characters of the P_DESCRIPT attribute use: COLUMN P_DESCRIPT FORMAT A12 TRUNCATE**

Task: Discuss the need for using the command COMMIT.

Single-table queries using SQL commands.

DATA MANIPULATION COMMANDS: These include: INSERT – to insert data into a table; SELECT – to query data in the database; UPDATE – to update data in a table; DELETE – to delete data from a table.

THE SELECT COMMAND

```
SELECT P_CODE, P_DESCRIPT, P_INDATE, P_QOH, P_MIN, P_PRICE, P_DISCOUNT, V_CODE
        FROM PRODUCT;
```

UPDATE COMMAND: Use the UPDATE command to modify data in a table. The syntax for this command is:

```
UPDATE tablename
        SET
                columnname = expression [, columnname = expression]
        [WHERE conditionlist ];
```

Example:change P_INDATE from December 13, 2011, to January 18, 2012, of the PRODUCT table, use the primary key (13-Q2/P2) to locate the correct row. Therefore, type:

```
UPDATE PRODUCT
        SET   P_INDATE = '18-JAN-2012'   WHERE P_CODE = '13-Q2/P2';
```

If more than one attribute is to be updated in the row, separate the corrections with commas:

```
UPDATE PRODUCT SET P_INDATE = '18-JAN-2012', P_PRICE = 17.99, P_MIN = 10 12', P_PRICE = 17.99
P_MIN = 10;
```

- In update command, do not specify WHERE.

Task: Discuss the need for using the command UPDATE.

Single-table queries using SQL commands.

DATA MANIPULATION COMMANDS: **These include: INSERT – to insert data into a table; SELECT – to query data in the database; UPDATE – to update data in a table; DELETE – to delete data from a table.**

RESTORING THE TABLE CONTENT – from insert and update

ROLLBACK command-ROLLBACK undoes any changes since the last COMMIT command and brings the data back to the values that existed before the changes were made. To restore the data to their "prechange" condition, type:

ROLLBACK; and then press Enter.

THE DELETE COMMAND

Removes a table row; use the DELETE and its syntax is:

```
DELETE FROM tablename
        [WHERE
                conditionlist ];
```

Example:

```
DELETE FROM PRODUCT   WHERE   P_CODE = 'BRT-345';
DELETE FROM PRODUCT  WHERE  P_MIN = 5;
```

NB. WHERE condition may be optional at times.

Task: Discuss the need for using the command COMMIT.

Single-table queries using SQL commands.

DATA MANIPULATION COMMANDS: **These include: INSERT – to insert data into a table; SELECT – to query data in the database; UPDATE – to update data in a table; DELETE – to delete data from a table.**

SELECTING ROWS WITH CONDITIONAL RESTRICTIONS

```
SELECT columnlist
FROM tablelist
[WHERE conditionlist ];
```

Examples:

```
SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE V_CODE = 21344;

SELECT P_DESCRIPT, P_INDATE, P_PRICE, V_CODE
FROM PRODUCT
WHERE V_CODE <> 21344;

SELECT P_DESCRIPT, P_QOH, P_MIN, P_PRICE
FROM PRODUCT
WHERE P_PRICE <= 10;
```

Task: **Formulate different queries using SELECT command**

Enforcing referential integrity using SQL.

- **Referential integrity refers to the relationship between tables.**
- **Each table in a database must have a primary key, this primary key can appear in other tables because of its relationship to data within those tables.**
- **When a primary key from one table appears in another table, it is called a foreign key.**
- **Foreign keys join tables and establish dependencies between tables.**
- **Tables can form a hierarchy of dependencies in such a way that if you change or delete a row in one table, you destroy the meaning of rows in other tables.**
- **For example, the the customer_num column of the customer table is a primary key for that table and a foreign key in the orders and cust_call tables.**
- **Customer number 300, Francis Wekulo, is referenced in both the orders and cust_calls tables.**
- **If customer 300 is deleted from the customer table, the link between the three tables and this particular customer is destroyed.**

Task: give examples of the various dependencies above.

Enforcing referential integrity using SQL.

- **The Relational Database Model,requires adherence to rules for entity integrity and referential integrity. Most SQL implementations support both inFrantegrity rules.**

- **Entity integrity is enforced automatically when the primary key is specified in the CREATE TABLE command sequence.**

- **For example, you can create the VENDOR table structure and set the stage for the enforcement of entity integrity rules by using the following: PRIMARY KEY (V_CODE)**

Task: give examples of the various dependencies above.

Enforcing referential integrity using SQL.

- In the PRODUCT table's CREATE TABLE sequence, note that referential integrity has been enforced by specifying the following in the PRODUCT table:

FOREIGN KEY (V_CODE) REFERENCES VENDOR ON UPDATE CASCADE

The foreign key constraint definition ensures that:

- You cannot delete a vendor from the VENDOR table if at least one product row references that vendor.
- This is the default behavior for the treatment of foreign keys.
- On the other hand, if a change is made in an existing VENDOR table's V_CODE, that change must be reflected automatically in any PRODUCT table V_CODE reference (ON UPDATE CASCADE).
- That restriction makes it impossible for a V_CODE value to exist in the PRODUCT table if it points to a nonexistent VENDOR table V_CODE value.
- In other words, the ON UPDATE CASCADE specification ensures the preservation of referential integrity.

Task: **Discuss the importance of referential integrity.**

Enforcing referential integrity using SQL.

- **Oracle does not support ON UPDATE CASCADE. In general, ANSI SQL permits the use of ON DELETE and ON UPDATE clauses to cover CASCADE, SET NULL, or SET DEFAULT.**

Note About Referential Constraint Actions

The support for the referential constraint's actions varies from product to product. For example:

- **MS Access, SQL Server, and Oracle support ON DELETE CASCADE.**
- **MS Access and SQL Server support ON UPDATE CASCADE.**
- **Oracle does not support ON UPDATE CASCADE.**
- **Oracle supports SET NULL.**

Task: Give examples of referential integrity.

The SQL: 1999 and SQL: 200n standards.

SQL: 1999 standard

- This part of ISO/IEC 9075 defines the data structures and basic operations on SQL-data. It provides functional capabilities for creating, accessing, maintaining, controlling, and protecting SQL-data.
- This part of ISO/IEC 9075 specifies the syntax and semantics of a database language:
  - For specifying and modifying the structure and the integrity constraints of SQL-data.
  - For declaring and invoking operations on SQL-data and cursors.
  - For declaring database language procedures.
- It also specifies an Information Schema that describes the structure and the integrity constraints of SQL-data.
- This part of ISO/IEC 9075 provides a vehicle for portability of data definitions and compilation units
- between SQL-implementations.
- This part of ISO/IEC 9075 provides a vehicle for interconnection of SQL-implementations. Implementations of this part of ISO/IEC 9075 may exist in environments that also support application programming languages, end-user query languages, report generator systems, data dictionary systems, program library systems, and distributed communication systems, as well as various tools for database design, data administration, and performance optimization.

Task: **Describe the various normal forms?**

The SQL: 1999 and SQL: 200n standards.

SQL:200n Standards

- In the 21st century, the SQL standard has been regularly updated.
- The SQL:2003 standard was published on March 1, 2004, it added the window functions, a powerful analytical feature that allows you to compute summary statistics without collapsing rows. They are extremely useful in preparing all kinds of business reports, analyzing time series data, and analyzing trends.
- The addition of window functions to the standard coincided with the popularity of OLAP and data warehouses.
- People started using databases to make data-driven business decisions.
- SQL:2003 also introduced XML-related functions, sequence generators, and identity columns.
- SQL:2006 further specified how to use SQL with XML addressed SQL-XML interoperability.
- SQL:2008 legalized the use of ORDER BY outside cursor definitions(!), and added INSTEAD OF triggers, the TRUNCATE statement, and the FETCH clause.
- SQL:2011 added temporal data and some enhancements to window functions and the FETCH clause.

Source: https://learnsql.com/blog/history-of-sql-standards/

Task: Discuss the evolution of the SQL standards.

The SQL: 1999 and SQL: 200n standards.

SQL:200n Standards

- **SQL:2016 added row pattern matching and polymorphic table functions as well as long awaited JSON support.**
- **In the 2010s, JSON replaced XML as the common data exchange format;**
- **Modern Internet applications use JSON instead of XML as their data format.**
- **The emerging NoSQL movement also popularized JSON;**
- **document databases store JSON files, and key-value stores are compatible with the JSON format.**
- **The SQL standard added JSON support to allow for interoperability with modern applications and new types of databases.**
- **The current SQL standard is SQL:2019. It added Part 15, which defines multidimensional array support in SQL.**

Source: https://learnsql.com/blog/history-of-sql-standards/

Task: Discuss the evolution of the SQL standards.

Week 5 exercises

1)Interpret the history and role of SQL in database development.
2)Define a database using the SQL data definition language.
3)Write single-table queries using SQL commands.
4)Establish referential integrity using SQL.
5)Discuss the SQL: 1999 and SQL: 200n standards.

Week 5 Session References

- **[Course Text] Carlos Coronel, Steven Morris, Peter Rob and Keeley Crockett Database Principles: Fundamentals of Design, Implementation, and Management, 14th Edition, 2022, ISBN-13978-0357673034.**

- **Thomas M. Connolly, Carolyn E. Begg (2021). Database Systems: A Practical Approach to Design, Implementation, and Management. Published by Pearson (July 14th 2021). ISBN-13: 9780137517053**

Thank You