Murach Chapter 10 Part 1

How to Work with Tables

Week 6

Knowledge Points in this lecture

- Create table basics
- Create table with Primary Key constraints
- Create table with Foreign Key constraints

The syntax of the CREATE TABLE statement

```
CREATE TABLE [schema_name.]table_name
(
  column_name_1 data_type [column_attributes]
  [, column_name_2 data_type [column_attributes]]...
  [, table_level_constraints]
)
```

Common column attributes

- NOT NULL – can not have NULL values
- UNIQUE – no duplicate values, but allows NULL values
- DEFAULT – specify a default value other than NULL

A statement that creates a table without column attributes

```
CREATE TABLE vendors
(
  vendor_id      NUMBER,
  vendor_name    VARCHAR2(50)
)
```

A statement that creates a table with column attributes

```
CREATE TABLE vendors
(
  vendor_id      NUMBER           NOT NULL     UNIQUE,
  vendor_name    VARCHAR2(50)     NOT NULL     UNIQUE
)
```

Another statement that creates a table with column attributes

```
CREATE TABLE invoices
(
  invoice_id      NUMBER         NOT NULL  UNIQUE,
  vendor_id       NUMBER         NOT NULL,
  invoice_number  VARCHAR2(50)   NOT NULL,
  invoice_date    DATE                     DEFAULT SYSDATE,
  invoice_total   NUMBER(9,2)    NOT NULL,
  payment_total   NUMBER(9,2)              DEFAULT 0
)
```

The syntax of a column-level primary key constraint

```
[CONSTRAINT constraint_name] PRIMARY KEY
```

The syntax of a table-level primary key constraint

```
[CONSTRAINT constraint_name]
PRIMARY KEY (column_name_1 [, column_name_2]...)
```

[A]: A is optional

Composite primary keys must be defined at table-level.

A table with column-level constraints

```
CREATE TABLE vendors
(
  vendor_id      NUMBER         PRIMARY KEY,
  vendor_name    VARCHAR2(50)   NOT NULL       UNIQUE
)
```

A table with named column-level constraints

```
CREATE TABLE vendors
(
  vendor_id      NUMBER
                 CONSTRAINT vendors_pk PRIMARY KEY,
  vendor_name    VARCHAR2(50)
                 CONSTRAINT vendor_name_nn NOT NULL
                 CONSTRAINT vendor_name_un UNIQUE
)
```

A table with table-level constraints

```
CREATE TABLE vendors
(
  vendor_id      NUMBER,
  vendor_name    VARCHAR2(50)     NOT NULL,
  CONSTRAINT vendors_pk PRIMARY KEY (vendor_id),
  CONSTRAINT vendor_name_uq UNIQUE (vendor_name)
)
```

A table with a two-column composite primary key constraint

```
CREATE TABLE invoice_line_items
(
  invoice_id               NUMBER         NOT NULL,
  invoice_sequence         NUMBER         NOT NULL,
  line_item_description    VARCHAR2(100)  NOT NULL,
  CONSTRAINT line_items_pk
      PRIMARY KEY (invoice_id, invoice_sequence)
)
```

Terms to know

- **Constraint** – integrity rules on data stored in a table column
- **Column-level constraint**
  - specified as part of the definition of the column it constrains
- **Table-level constraint**
  - specified separately from column definition
- **Not null constraint** – no NULL values
- **Unique constraint** – no duplicate values; NULL values: yes
- **Primary key constraint** – same as (NOT NULL + UNIQUE)
- **Foreign key constraint – for referential integrity**
  - Requires: value in one table to match value in another table
  - Allows NULL values
- **Check constraint** – check condition on data stored in a table column

The syntax of a column-level foreign key constraint

```
[CONSTRAINT constraint_name]
  REFERENCES table_name (column_name)
  [ON DELETE {CASCADE|SET NULL}]
```

The syntax of a table-level foreign key constraint

```
[CONSTRAINT constraint_name]
  FOREIGN KEY (column_name_1 [, column_name_2]...)
  REFERENCES table_name (column_name_1
                         [, column_name_2]...)
  [ON DELETE {CASCADE|SET NULL}]
```

[A]: A is optional
{A | B}: choice of A or B

A table with a column-level foreign key constraint

```
CREATE TABLE invoices
(
  invoice_id      NUMBER    PRIMARY KEY,
  vendor_id       NUMBER    REFERENCES vendors (vendor_id),
  invoice_number VARCHAR2(50) NOT NULL     UNIQUE
)
```

A table with a table-level foreign key constraint

```
CREATE TABLE invoices
(
  invoice_id        NUMBER           NOT NULL,
  vendor_id         NUMBER           NOT NULL,
  invoice_number    VARCHAR2(50)     NOT NULL    UNIQUE,
  CONSTRAINT invoices_pk
    PRIMARY KEY (invoice_id),
  CONSTRAINT invoices_fk_vendors
    FOREIGN KEY (vendor_id)
        REFERENCES vendors (vendor_id)
)
```