

Murach Chapter 10 Part 2

How to Work with Tables

Week 8, Lecture 15,
Spring 2019

Knowledge Points in this lecture

- Change column definitions using ALTER TABLE
- Change the constraints using ALTER TABLE
- Rename a table
- Truncate a table
- Drop a table
- Script to create tables in AP user
- View columns and constraints in a table in SQL Developer

The syntax for modifying the columns of a table

```
ALTER TABLE [schema_name.]table_name
{
  ADD          column_name data_type [column_attributes] |
  DROP COLUMN column_name |
  MODIFY       column_name data_type [column_attributes]
}
```

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

A statement that adds a new column

```
ALTER TABLE vendors  
ADD last_transaction_date DATE;
```

A statement that drops a column

```
ALTER TABLE vendors  
DROP COLUMN last_transaction_date;
```

Cannot drop certain columns.

- Example: primary key column(s)

A statement that changes the **length** of a column

```
ALTER TABLE vendors  
MODIFY vendor_name VARCHAR2(100); --old: VARCHAR2(50)
```

A statement that changes the **type** of a column

```
ALTER TABLE vendors  
MODIFY vendor_name CHAR(100); --old: VARCHAR2
```

A statement that changes a default value

```
ALTER TABLE vendors  
MODIFY vendor_name DEFAULT 'New Vendor';
```

A statement that fails because it would lose data

```
ALTER TABLE vendors  
MODIFY vendor_name VARCHAR2(10); --old: VARCHAR2(50)
```

The response from the system

```
SQL Error: ORA-01441: cannot decrease column  
length because some value is too big
```

Warning

- You should never alter a table or other database object in a **production database** without consulting the DBA.

The syntax for modifying the constraints of a table

```
ALTER TABLE table_name
{
  ADD          CONSTRAINT  constraint_name
                  constraint_definition [DISABLE] |
  DROP          CONSTRAINT  constraint_name |
  ENABLE [NOVALIDATE] constraint_name |
  DISABLE          constraint_name
}
```

Constraint status:

- Disabled - does not apply to (existing and new) data.
- Enabled no-validated – apply only to new data
- Enabled validated – apply to both new and existing data
 - Default status for a new constraint
 - If existing data violates the constraint, its status canNOT be changed to “Enabled validated”

A statement that adds a new check constraint

```
ALTER TABLE invoices  
ADD CONSTRAINT invoice_total_ck  
CHECK (invoice_total >= 0);
```

A statement that drops a check constraint

```
ALTER TABLE invoices  
DROP CONSTRAINT invoice_total_ck;
```

A statement that adds a disabled constraint

```
ALTER TABLE invoices  
ADD CONSTRAINT invoice_total_ck  
CHECK (invoice_total >= 1) DISABLE;
```

A statement that enables a constraint for new values only

```
ALTER TABLE invoices  
ENABLE NOVALIDATE CONSTRAINT invoice_total_ck;
```

A statement that disables a constraint

```
ALTER TABLE invoices  
DISABLE CONSTRAINT invoice_total_ck;
```

A statement that adds a foreign key constraint

```
ALTER TABLE invoices
ADD CONSTRAINT invoices_fk_vendors
FOREIGN KEY (vendor_id) REFERENCES vendors (vendor_id);
```

A statement that adds a unique constraint

```
ALTER TABLE vendors
ADD CONSTRAINT vendors_vendor_name_uq
UNIQUE (vendor_name);
```

A statement that adds a not null constraint

```
ALTER TABLE vendors
MODIFY vendor_name
CONSTRAINT vendors_vendor_name_nn NOT NULL;
```

How Oracle handles new constraints

- By default, Oracle verifies that existing data satisfies a new constraint.
- If that's not what you want, you can add a disabled constraint or enabled but novalidated constraint.

A statement that renames a table

```
RENAME vendors TO vendor
```

A statement that deletes all data from a table

```
TRUNCATE TABLE vendor
```

A statement that deletes a table from the current schema

```
DROP TABLE vendor
```

A statement that qualifies the table to be deleted

```
DROP TABLE ex.vendor
```

DROP TABLE, TRUNCATE TABLE, DELETE

- DELETE FROM tablename
 - Remove only the data in the named table
- TRUNCATE TABLE tablename
 - Remove all data in the named table
 - Shrink the storage space
- DROP TABLE tablename
 - Remove all data in the named table
 - Release all storage space allocated to the named table
 - Remove the definition of the named table

Notes for creating tables in a script

- You must create the tables that don't have foreign keys first.
 - First create tables that are parent tables on which other tables (child tables) depend through foreign key constraints.
 - Tables that are not related to any other table can be created in any order.
- When you drop tables, you start by dropping the last table that was created and then work back to the first table that was created.
 - Drop child tables before parent tables

The script that creates the AP schema

```
CREATE TABLE general_ledger_accounts
(
    account_number          NUMBER          NOT NULL,
    account_description     VARCHAR2(50)    NOT NULL,
    CONSTRAINT gl_accounts_pk
        PRIMARY KEY (account_number),
    CONSTRAINT gl_account_description_uq
        UNIQUE (account_description)
);
```


The script that creates the AP schema (continued)

```
CREATE TABLE terms
(
    terms_id                NUMBER                NOT NULL,
    terms_description        VARCHAR2(50)         NOT NULL,
    terms_due_days           NUMBER                NOT NULL,
    CONSTRAINT terms_pk
        PRIMARY KEY (terms_id)
);
```

The script that creates the AP schema (continued)

```
CREATE TABLE vendors
(
    vendor_id                NUMBER                NOT NULL,
    vendor_name              VARCHAR2( 50 )        NOT NULL,
    vendor_address1          VARCHAR2( 50 ),
    vendor_address2          VARCHAR2( 50 ),
    vendor_city              VARCHAR2( 50 )        NOT NULL,
    vendor_state             CHAR( 2 )            NOT NULL,
    vendor_zip_code          VARCHAR2( 20 )        NOT NULL,
    vendor_phone             VARCHAR2( 50 ),
    vendor_contact_last_name VARCHAR2( 50 ),
    vendor_contact_first_name VARCHAR2( 50 ),
```

The script that creates the AP schema (continued)

```
default_terms_id          NUMBER          NOT NULL,  
default_account_number    NUMBER          NOT NULL,  
CONSTRAINT vendors_pk  
    PRIMARY KEY (vendor_id),  
CONSTRAINT vendors_vendor_name_uq  
    UNIQUE (vendor_name),  
CONSTRAINT vendors_fk_terms  
    FOREIGN KEY (default_terms_id)  
    REFERENCES terms (terms_id),  
CONSTRAINT vendors_fk_accounts  
    FOREIGN KEY (default_account_number)  
    REFERENCES general_ledger_accounts (account_number)  
);
```

Table vendors is a child table of table terms and table general_ledger_accounts

The script that creates the AP schema (continued)

```
CREATE TABLE invoices
(
    invoice_id          NUMBER          NOT NULL,
    vendor_id           NUMBER          NOT NULL,
    invoice_number       VARCHAR2(50)   NOT NULL,
    invoice_date         DATE           NOT NULL,
    invoice_total        NUMBER(9,2)    NOT NULL,
    payment_total        NUMBER(9,2)    DEFAULT 0,
    credit_total         NUMBER(9,2)    DEFAULT 0,
    terms_id            NUMBER          NOT NULL,
    invoice_due_date     DATE           NOT NULL,
    payment_date         DATE,
```

The script that creates the AP schema (continued)

```
CONSTRAINT invoices_pk  
    PRIMARY KEY (invoice_id),  
CONSTRAINT invoices_fk_vendors  
    FOREIGN KEY (vendor_id)  
    REFERENCES vendors (vendor_id),  
CONSTRAINT invoices_fk_terms  
    FOREIGN KEY (terms_id)  
    REFERENCES terms (terms_id)  
);
```

Table invoices is a child table of table vendors and table terms

The script that creates the AP schema (continued)

```
CREATE TABLE invoice_line_items
(
    invoice_id          NUMBER          NOT NULL,
    invoice_sequence    NUMBER          NOT NULL,
    account_number      NUMBER          NOT NULL,
    line_item_amt       NUMBER(9,2)     NOT NULL,
    line_item_description VARCHAR2(100) NOT NULL,
    CONSTRAINT line_items_pk
        PRIMARY KEY (invoice_id, invoice_sequence),
    CONSTRAINT line_items_fk_invoices
        FOREIGN KEY (invoice_id)
            REFERENCES invoices (invoice_id),
    CONSTRAINT line_items_fk_accounts
        FOREIGN KEY (account_number)
            REFERENCES general_ledger_accounts (account_number)
);
```

Table invoice_line_items is a child table of table invoices and table general_ledger_accounts

The column definitions for the Invoices table

The screenshot displays the Oracle SQL Developer interface. The title bar reads "Oracle SQL Developer : Table AP.INVOICES@ap". The menu bar includes File, Edit, View, Navigate, Run, Team, Tools, Window, and Help. The toolbar contains icons for file operations, navigation, and execution. On the left, the "Connections" pane shows a tree view with "ap" selected, containing various database objects like Tables, Views, and Indexes. The "Tables (Filtered)" section lists several tables, with "INVOICES" highlighted. The main pane shows the "Columns" tab for the "INVOICES" table, displaying a list of 10 columns with their respective data types, nullability, and default values. The "Columns" tab is active, showing the following data:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	INVOICE_ID	NUMBER	No	(null)	1	(null)
2	VENDOR_ID	NUMBER	No	(null)	2	(null)
3	INVOICE_NUMBER	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4	INVOICE_DATE	DATE	No	(null)	4	(null)
5	INVOICE_TOTAL	NUMBER(9,2)	No	(null)	5	(null)
6	PAYMENT_TOTAL	NUMBER(9,2)	Yes	0	6	(null)
7	CREDIT_TOTAL	NUMBER(9,2)	Yes	0	7	(null)
8	TERMS_ID	NUMBER	No	(null)	8	(null)
9	INVOICE_DUE_DATE	DATE	No	(null)	9	(null)
10	PAYMENT_DATE	DATE	Yes	(null)	10	(null)

The bottom status bar shows "ap | AP | INVOICE".

The constraints for the Invoices table

The screenshot displays the Oracle SQL Developer interface. The title bar indicates the current context is 'Table AP.INVOICES@ap'. The main window is divided into several panes. On the left, the 'Connections' pane shows a tree view of database objects for the 'ap' connection, with 'INVOICES' selected under the 'Tables (Filtered)' folder. The central pane shows the 'Constraints' tab for the 'INVOICES' table. It contains a table listing eight constraints, including two foreign keys and six check constraints. Below this, the 'Columns' tab is visible, showing a list of columns with placeholder names 'COLUMN_...'. The bottom status bar shows the path 'ap | AP | INVOICE'.

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE
1	INVOICES_FK_TERMS	Foreign_Key	(null)	AP	TERMS
2	INVOICES_FK_VENDORS	Foreign_Key	(null)	AP	VENDORS
3	INVOICES_PK	Primary_Key	(null)	(null)	(null)
4	SYS_C007231	Check	"VENDOR_ID" IS NOT NULL	(null)	(null)
5	SYS_C007232	Check	"INVOICE_NUMBER" IS NOT NULL	(null)	(null)
6	SYS_C007233	Check	"INVOICE_DATE" IS NOT NULL	(null)	(null)
7	SYS_C007234	Check	"INVOICE_TOTAL" IS NOT NULL	(null)	(null)
8	SYS_C007235	Check	"TERMS_ID" IS NOT NULL	(null)	(null)