Data Management and Database Design

DMDD 6210

Week #7

Northeastern University

This week

- Database Scaling
- Data Definition Language
- Data Manipulation Language

Quiz

```
SQL> DESC LEAD
 Name
 ----------------------
 COMPANY       VARCHAR2(30)
 PHONE         VARCHAR2(20)

SQL> select * from lead;
COMPANY                       PHONE
------------------------ ---------------
Acme Production               5553214321
Basic Apparel          002495559875432
Century Movies         456123789
Danish Design          004566554433
Ewok Emporium          86427531
```

Phone number has to be used as account ID in CUSTOMER Table.

Challenge is –

account ID is a VARCHAR2(10) in customer table

Expected output –

```
ACCOUNTID COMPANY
---------- ------------------------
456123789  Century Movies
4566554433 Danish Design
5553214321 Acme Production
5559875432 Basic Apparel
86427531   Ewok Emporium
```

Quiz

- **Clue:**
  - Use **SUBSTR** function
  - Use **GREATEST** function

```
select company, substr(phone, greatest(-10,-length(phone)), 10) as account_id
from lead;
ACCOUNTID COMPANY
---------- ---------------
456123789  Century Movies
4566554433 Danish Design
5553214321 Acme Production
5559875432 Basic Apparel
86427531   Ewok Emporium
```

Scaling

What is Horizontal and Vertical scaling?

What is auto scaling?

Data Definition Language

- DDL is subset of SQL commands to CREATE / MODIFY / DELETE objects
- These commands have immediate effect on DB records such information to –
  - **DATA DICTIONARY (Aka METADATA / CATALOG)**
- As we all know tables are created using CREATE TABLE command
  - Tables are owned by user who creates it
  - Table names within that user schema are unique
  - Column names within a table must be unique
- Table naming conventions
  - Must begin with an Alphabet
  - Can contain numbers and special characters such as $, _
  - 30 characters in length
  - Not case sensitive (However we can force it to be case sensitive and is not recommended)

Column Constraints

- NOT NULL – Blocks NULL values
- UNIQUE – Maintains unique values in that column however accepts NULLs
- PRIMARY KEY – Maintains unique values and blocks NULL values
- CHECK – Allows to specify condition to check on each row
- DEFAULT – Assigns default value for the column
- REFERENCES – Foreign key constraint to maintain Referential Integrity

Column Constraints

Create a table called ITEMMAST and lets discuss the way constraints defined.

```
CREATE TABLE itemmast (
  item_no         number(4)        PRIMARY KEY,
  name            varchar2(20)     NOT NULL constraint ITM_NN unique,
  category        varchar(1)       check (category in (’A’,’B’,’C’)),
  qoh             number(5)        default 50,
  uom             char(4)          check (uom > 0),
  rate            number(8,2)      not null
);
```

Lets make the logic bit complex, Include checks on RATE based on Category as below –

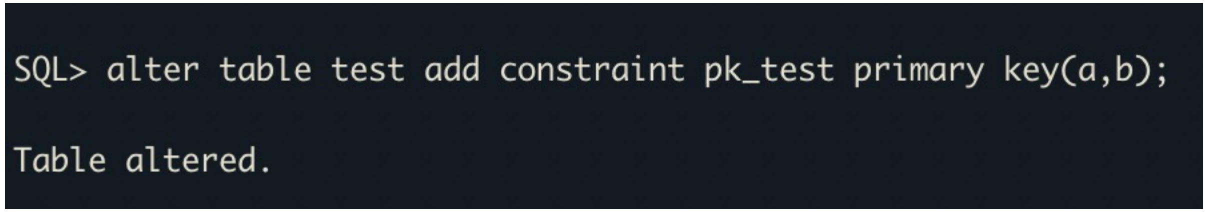
- If CATEGORY = ‘A’ then Rage should be < $1000
- If CATEGORY = ‘B’ then Rage should be > $1000 and <$4500
- If CATEGORY = ‘C’ then Rage should be >$4500

Column Constraints

```
CREATE TABLE itemmast (
  item_no    number(4)     PRIMARY KEY,
  name       varchar2(20) NOT NULL constraint ITM_NN unique,
  category   varchar(1)    check (category in ('A','B','C')),
  qoh        number(5)     default 50,
  uom        char(4)       check (uom > 0),
  rate       number(8,2)   not null,
  CHECK ((category = 'A' and RATE <1000) OR (category = 'B' and RATE >100
  and RATE <4500) OR (category = 'C' and RATE >4500))
);
```

Table Level constraints

- Column level constraint scope only column
- Table level constraint can see every column which gives option to assign any column or columns to this constraint.
- Every Column level constraint can be expressed at the table level except NOT NULL
- Primary key → 2 types



```
SQL> alter table test add constraint pk_test primary key(a,b);

Table altered.
```

  - Simple Primary Key
    - `Alter table emp add constraint emp_pk primary key (empno);`
    - `Create table product_mast(id number constraint product_mast_pk primary key, ….. );`
  - Composite Primary Key

Points to remember for Primary Key

- Primary key cannot be deleted if reference exists
- A table can have only ONE primary key
- Composite primary key – Combination will be unique
- It is not mandatory to have Primary key however, Its recommended to have it.

- **Note** – Primary Key data Rows cannot be deleted until Referenced data is deleted from child tables.

Create Entity

Create a table called ITEMTRAN to show the references constraint

```
    CREATE TABLE itemtran (
      Id                    number primary key,
      Item_no        number references itemmast(item_no),
      Tran_type             varchar2(2) check (tran_type in (‘CC’,’CH’),
      Tran_Dt        date,
      Qty                   number(5)
    );
```

Create Entity

Create a table called ITEMTRAN to show the references constraint

```
CREATE TABLE itemtran (
  Id number primary key,
  Item_no number references itemmast(item_no),
  Tran_type varchar2(2) check (tran_type in (‘CC’,’CH’),
  Tran_Dt date,
  Qty number(5)
);
```

How to Rename a column / Table?

Lets say we created a table with wrong spelling or for some reasons we need to change the column name. In the same way this can happen for Table name as well.

Rename Entity/Attribute

- **ALTER TABLE** <TABLE NAME> **RENAME TO** <NEW NAME>
- **ALTER TABLE** <TABLE NAME> **RENAME COLUMN** <COLUMN NAME> **TO** <NEW COLUMN NAME>

- ADD a column to an existing table then we can use below –
  ALTER TABLE <table-name> ADD(<column-name> <datatype>);

- ADD constraint to existing column –

```
    ALTER TABLE <table-name>
    ADD CONSTRAINT <constraint-name> (
        <Column Name>
        );
```

- Modify Datatype length –
  ALTER TABLE <table-name> MODIFY(<column-name> <datatype>(length));

Enable / Disable Constraints

ALTER TABLE <TABLE NAME> **DISABLE CONSTRAINT** <CONSTRAINT NAME>

ALTER TABLE <TABLE NAME> **ENABLE CONSTRAINT** <CONSTRAINT NAME>

How to Delete an Entity ?

DROP TABLE <table-name>

Does drop has any impact on dependency?

Lets check with examples via Demo

On Delete Cascade

When a PARENT row is Deleted, all the corresponding CHILD rows are deleted.

This option is always used in conjunction with FOREIGH KEY

```
CREATE TABLE itemtran (
   Id number primary key,
   Item_no number references itemmast(item_no) ON DELETE CASCADE,
   Tran_type varchar2(2) check (tran_type in (‘CC’,’CH’),
   Tran_Dt date,
   Qty number(5)
);
```

Note: One more way of creating a table is using a SELECT statement as below –

```
CREATE TABLE emp_temp as SELECT * from emp;
```

On Delete Set NULL

Sets all the records of the column which is defined as a foreign key in the child table to Null

if the corresponding record in the parent table is deleted.

how to define an "On Delete Set Null" clause with foreign key?

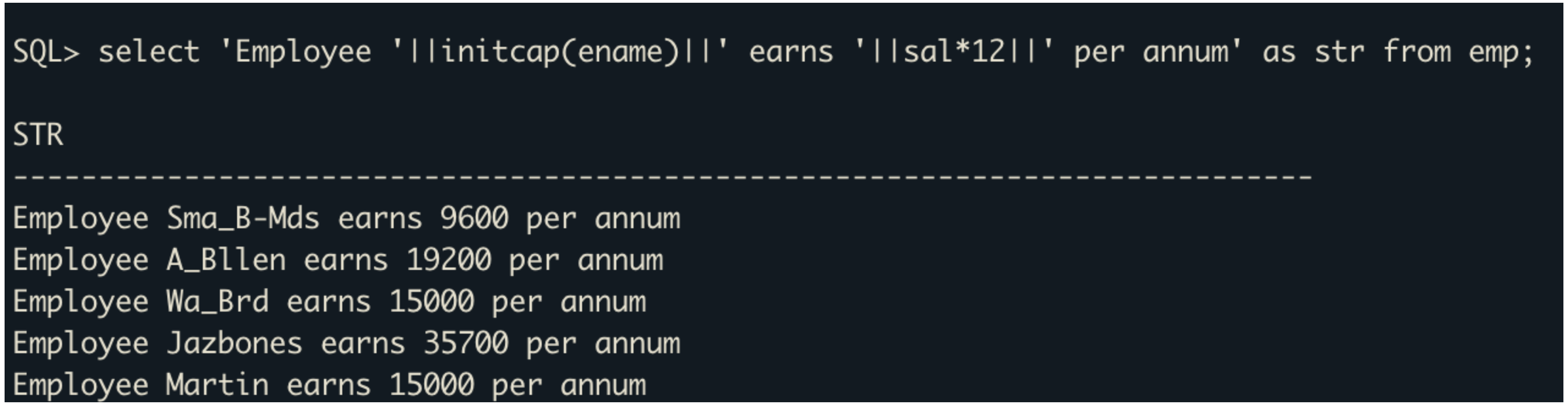
using key words "**`on delete set null`**" similar to **`ON DELETE CASCADE`**

Catalog

- To view list of all objects in a schema
  - SELECT * FROM CAT; -- Catalog

String concatenation using Pipe symbol

- Note: In Oracle Data can be concatenated using two pipe symbols → ||
  - **Example: select empno||’ – ‘||ename from emp;**
  - Another example see below screen shot –



```
SQL> select 'Employee '||initcap(ename)||' earns '||sal*12||' per annum' as str from emp;

STR
---------------------------------------------------------------------------
Employee Sma_B-Mds earns 9600 per annum
Employee A_Bllen earns 19200 per annum
Employee Wa_Brd earns 15000 per annum
Employee Jazbones earns 35700 per annum
Employee Martin earns 15000 per annum
```

  - Note: We can use this concatenation in Insert statements as well.

Questions?