

Comp 4150: Advanced and Practical Database Systems

More on SQL Features

Character and Number Functions

- Text literals are always surrounded by single quotes

1. LOWER function:

Transforms data into lower case eg.

```
Select state , LOWER(state), LOWER('State')
```

```
From zipcode
```

2. The LPAD and RPAD Functions:

These are the leftpad and rightpad strings for padding a string with specified characters, e.g

```
Select RPAD(city,20,'*') "City Name", LPAD(state,10,'-') "State Name"
```

```
From zipcode
```

Syntax of LPAD and RPAD functions is

```
LPAD(char1,n[,char2])
```

```
RPAD(char1,n[,char2])
```

3. The SUBSTR function:

E.g (Display 5 characters from position 1 of string)

```
Select last_name ,  
       SUBSTR (last_name ,1,5),  
       SUBSTR (last_name,6)
```

```
From student
```

Second displays the rest of the string from 6th character. Syntax is:

```
SUBSTR (char1, n [,char2] )
```

4. INSTR Function:

Meaning in string looks for the occurrence of a string inside another string and returns the number of the position where the string starts. E.g

```
Select description , INSTR ( description, 'er')
```

```
from course
```

- Syntax is

```
INSTR ( char1, char2 [,n1[,n2]] )
```

- Third parameter specifies the start position of the string for search and fourth is which string to look for.

5. The Dual Table:

Contains a single row and column called 'dummy' and returns a single row. It can be used with functions to select values not stored in tables such as today's date, e.g.

Select length('Hello There')
from dual

6. LENGTH('String'): returns number of characters

- Syntax

LENGTH (String)

7. Concatenation (||) : Can also be done with Concat functions. E.g.

Select CONCAT (city , state)

From zipcode

Syntax: concat (par1 , par2)

CONCAT function takes only 2 parameters. Using || to concat several string. E.g.

Select city || ',' || state || ' ' || zip

From zipcode

8. TRANSLATE function: e.g:

Select phone

from student

where TRANSLATE (phone, '0123456789-', '#####-') <> '###-###-####'

Syntax:

TRANSLATE (char, if , then)

9. The REPLACE Function

E.g

Select REPLACE ('My hand is asleep', 'hand','foot') from dual

- Syntax

REPLACE (char, If, then)

10. LTRIM and RTRIM functions:

Left trim and right trim string by deleting second parameter. E.g

Select zip, LTRIM (zip,'0'), RTRIM(zip,'4')

from zipcode

order by zip

- Syntax

LTRIM (string [,character to delete])

RTRIM (string [,character to delete])

If second parameter is not specified it trims spaces.

11. UPPER function. (Transforms string to upper case). E.g

select state UPPER(state), UPPER('State') from zipcode

- Syntax UPPER(string)

12. INITCAP function: Capitalizes first letter of a word and the rest to lower case.

E.g

Select description "Description", INITCAP (description) "Initcap Description"

from course

where description like '%SQL'
- Syntax
 INITCAP (string)

Number Functions

The ABS function.

E.g

Select 'The absolute value of -29 is ' || ABS(-29) from dual

- Syntax ABS(value)

The SIGN function:

Returns 1 if sign is +ve, -1 for -ve and 0 for zero

select -14, sign(-14) , sign (14), sign(0),ABS(-14) from dual

- Syntax SIGN(value)

MOD function.

E.g

select MOD(20,7) from dual

Syntax: MOD(value, divisor)

ROUND versus TRUNC.

E.g

Selet 222,34501, round(222.34501 , 2) , trunk (222.34501, 2)
from dual

Syntax: ROUND (value[, precision])

 TRUNC (value[, precision])

Date and Conversion Functions

TO_NUMBER:

Converts a VARCHAR2 or CHAR to a NUMBER

Select course_no, description

from course

where course_no = TO_NUMBER ('350')

TO_CHAR:

Converts a DATE or NUMBER to a VARCHAR2 . E.g.

select course_no , cost, to_char(cost, '999,999') formatted from course

where course_no < 25

TO_DATE:

Converts a VARCHAR2, CHAR or NUMBER to a DATE . E.g

Select course_no, cost, TO_CHAR(cost,' 999,999') formatted from course

where course_no < 25

The above SQL achieves the same as the sqlplus instruction

COL cost FORMAT 999,999

Common number formats

Format Mask	Example	Result
999,999.99	.45	0.45
\$99,999.99	1234	\$1234.00
999	123.59	124

Applying Oracle's Date format model

TO_CHAR:

- Oracle DATE datatype consists of a date and time stamp. It displays date in default format DD-MON-YY format as 23-FEB-00

- To change to MM/DD/YYYY format use TO_CHAR as in

```
Select last_name, registration_date, TO_CHAR( registration_date, 'MM/DD/YYYY')  
from student
```

```
where registration_date = '22-JAN-99'
```

TO_DATE :

- Use to convert text literals as '22-JAN-99' to a standard date format as in

```
Select last_name , registration_date TO_CHAR ( registration_date, 'MM/DD/YYYY')  
from student
```

```
where registration_date = TO_DATE('01/22/1999','MM/DD/YYYY')
```

Date format mask is in p100.

The SYSDATE :

It returns the operating system's current date and time

```
Select sysdate from dual
```

ADD_MONTHS (date, integer)

LAST_DAY (date)

MONTHS_BETWEEN (date2, date1)

new_time (date, current time , zone, new time zone)

NEXT_DAY (date, day of the week)

Other Conversion functions

NVL :

Replaces a NULL value with a default values. As in

Select 60+60+NVL(BULL,1000) from dual

- Return 1120

-Syntax

NVL (input_expression, substitution_expr)

- The two parameters must be the same type.

DECODE

- Syntax

DECODE (if_expr, equals_search then_result [,else default])

E.g

Select distinct state,

DECODE (state,'NY', "New York",'NJ',"New Jersey") no default,

DECODE (state,'NY', "New York",'NJ',"New Jersey",'OTHER') with default

from zipcode

where state IN ('NY','NJ','CT')

The Group By and Having Clauses

- Aggregate functions are used with other columns in a SELECT list to tell the database you want the information grouped by other columns
- The group by clause must be used in this case to present the information in the output. E.g
Select location from section
group by location
- See p-137 for distinction between group_by and order_by
- The EXISTS operator returns either true or false, never unknown and tests only if a row exists using text literals like '1' or 'X' .E.g
Select instructor_id, last_name , first_name,zip from instructor i
where EXISTS
 (select 'x' from section
 where i.instructor_id = instructor_id)
- The above displays instructors where the instructor_id has a matching row in the section table.
- The OUTER join operator '(+)' is used to indicate that NULLS are generated for no matching rows.

- An example .
 Select c.course_no, s.course_no, s.section_id, c.description, s.start_date_time
 from course c , section s
 where c.course_no = s.course_no(+)
 UNION
 Select c.course_no, s.course_no, s.section_id, c.description, s.start_date_time
 from course c, section s
 where c.course_no (+) = s.course_no
- The first select statement performs and OUTER join on section table and the second select statement performs an outer join on the course table.
- **SAVEPOINT** command allows you to save the result of the DML transactions temporarily.
- **ROLLBACK** command can then refer back to a particular SAVEPOINT and roll back the transaction up to that point.
 E.g
 SAVEPOINT zip222
 ROLLBACK TO SAVEPOINT zip222
 The **truncate** command deletes all rows from a table but does not allow where clause. It automatically issues a COMMIT. E.g
 TRUNCATE TABLE class
- DDL commands can not be rolled back and they COMMIT any previously issued DML statements.

Common Data Types

	Data type	Use
1	VARCHAR2	For character data and requires column length up to 4000 chars
2	CHAR	Requires fixed column length up to 2000 chars
3	LONG	Stores up to 2 gigabytes of data
4	CLOB	Stores up to 4 gigabytes of data
5	NUMBER	Can store up to 38 decimal digits of precision. It does not require a precision and scale necessarily. Used to specify real value as e.g NUMBER (5,2) for numbers, -999.99 to 999.99
6	DATE	Stores the century, year, month, day, hour, minute and second in internal format, can be displayed using different format masks.
7	BLOBS, RAW, LONG RAW, BFILE	Each of these data types points to a binary operating system files.

Constraints

- Integrity constraints enforce business rules of a system. They ensure data integrity and data consistency among all applications.
- Constraints types include primary key constraints not NULL check constraints, foreign key constraints, the unique constraints, the check constraint, the data default. Eg.
Create table tabl (
col1 VARCHAR2(10) PRIMARY KEY,
col2 NUMBER(4) NOT NULL,
col3 VARCHAR2(5) REFERENCES zipcode(zip) ON DELETE CASCADE,
col4 DATE DEFAULT SYSDATE,
col5 NUMBER(2) UNIQUE,
col6 NUMBER CHECK(col6 < 100),
col7 VARCHAR2(5) UNIQUE))
- **Primary key** constraints on a column ensures all values in this column are NOT NULL and unique.
- The **NOT NULL** constraint rejects insertion of NULL values
- **FOREIGN Key** constraint defines domain of foreign key as that of the primary key that it relates to.
- **UNIQUE** constraint ensures every value is unique
- **CHECK** constraint ensures values meet specified condition
- **DATA DEFAULT** specifies a default option.

Naming Constrains

- Constraints can be named as in the following.
- Create table tabl (
col1 VARCHAR2(10),
col2 NUMBER(4) NOT NULL,
col3 NUMBER(6) ,
col4 DATE DEFAULT SYSDATE,
col5 VARCHAR2(5),
col6 NUMBER ,
col7 VARCHAR2(5)
CONSTRAINT tab1_pk PRIMARY KEY (col1)
CONSTRAINT tab1_zipcode_fk FOREIGN KEY (col13) REFERENCES
zipcode(zip),
CONSTRAINT tab1_uk UNIQUE (col5 , col6),
CONSTRAINT tabl1_col6_chk CHECK (cl6 < 100))
- Tables can be renamed with RENAME command. E.g
RENAME jan_99_enrollment to jan_99
- Tables can be dropped as
DROP TABLE jan_99

Altering Tables

- The characteristic of an already created table can be changed using the ALTER TABLE command in conjunction with the ADD, DROP and MODIFY clauses. E.g
ALTER TABLE toy
ADD (manufacturer VARCHAR2(30) NOT NULL)
or
Alter table TABLE toy
ADD (manufacturer VARCHAR2(30)
constraint toy_maun_nn NOT NULL)
- Constraints can be added, dropped, disabled or enabled through the Alter table command. E.g
ALTER TABLE Toy
ADD PRIMARY KEY (description)
- Alter TABLE toy
DROP CONSTRAINT toy_pk

Pseudocolumns

- Every row in the database has a unique address called ROWID which determines where the row is located
- Other oracle pseudocolumns are NEXTVAL, CURRVAL, ROWNUM and LEVEL.

INDEXES

- To optimize performance, you store indexes in separate tablespaces residing on different physical drives.
- To create an index and place it in a tablespace called index_tx, do the following
CREATE INDEX sect_location_i
ON section (location)
TABLESPACE index_tx
- Periodically, the Oracle B-tree index needs to be re-built as:
- ALTER INDEX stu_zip_fk_i REBUILD

SEQUENCES

- Sequences are Oracle database objects for generating unique integers.
- Sequences are typically incremented by 1 although increments are possible.
- To create a sequence named student_id_seq_new, do:
CREATE SEQUENCE student_id_seq_new
start with 1 NOCACHE
- The NOCACHE option prevents losing cached numbers if there is system crash as it disallows caching of sequences .
- To increment and sequence unique number, use NEXTVAL pseudocolumn
Select student_id_seq_new.NEXTVAL
from dual
- NEXTVAL is used in INSERT and UPDATE statements and to display the current value of the sequence after it is incremented use the CURRVAL pseudocolumn.

The data dictionary and Dynamic SQL Scripts

- Data dictionary is a set of tables and views containing data about the database.
- It is also called the catalog
- Dynamic SQL scripts simplify the writing of SQL Statement and the administration of the database.
- The scripts execute other SQL statements.
- The data dictionary has two sets of views:
the static data dictionary views and the dynamic views (also called v\$tables)
- Most fo the static views begin with the prefix user_, all_ or dba_
- The user_ views belong to the user querying the data dictionary
- The all_ views show the same information plus any information granted to the user by another user. It also shows public objects.
- The dba_ views show all the objects but you need DBA privileges to be able to query these views.
- The dynamic views begin with the V\$ and are used by the DBA to monitor the system.
- The views are dynamic because they are continuously updated by the database but never by the user.
- Data about static and dynamic data dictionary tables and views are contained in a view called dictionary , also called dict (synonym).

- We can find about it as
DESC dict
e.g. To find information about the sequences in the database, you can query the dict view as
select table_name, comments
from dict where table_name like '%SEQ%'
- There are four data dictionary views describing views (e.g SEQ)
- E.g Querying of the dictionary are
Select objects_name created, last_ddl_time, statuc from user_objects
- Select Column_name, comments from dict_columns
where column_name IN ('STATUS','LAST_DDL_TIME','CREATED')
- SELECT table_name from user_tables

Select table_name, column_name, data_type, data_length, nullable, data_default
from user_tab_columns
where table_name = 'GRADE'

Select constraint_name, last_number
from user_sequences
Select constraint_name, table_name,constraint_type from user_constraints
Select username from all_users

Dynamic SQL Scripts

- E.g & substitution variable for entering input at the keyboard as in.
select course_no, description from course
where course_no = &v_course_no
- Dynamic SQL allows you to execute SQL commands built at runtime
- Example
Select 'ALTER VIEW' || object_name || 'COMPILE;'
from user_objects
where object_type = 'VIEW'
AND status <> 'VALID'
- Others: Type in a .sql file and execute
PROMPT determines the maximum and min column
ACCEPT vcol CHAR PROMPT 'Enter the column name'
ACCEPT vtable CHAR PROMPT 'ENTER the table name'
SET verify OFF
Select MIN(&vcol), MAX(&vcol)
from &vtable

Exercise

- Explain each line in the following SQL script and describe the purpose of the script in one sentence [Ans. On p386-388]

- A single quote is used to enclose a text literal.

User Privileges, Roles and synonyms

- To establish a connection to the database using sqlplus, the user must be granted certain privileges.
- A user name is created with the CREATE USER command
- To create a new user first login as the user SYSTEM, default password is manager
- You can login with the CONNECT(CONN) command with parameter userid/password as in.
CONN system/manager
- You can also include host database string as in
CONN system/manager@scratchy
- The host string is found in the tnsnames.ora file listing the server name and the database instance name.
- The following creates a user MUSIC with password LISTEN
CREATE user music IDENTIFIED By listen
DEFAULT TABLESPACE user_data
TEMPORARY TABLESPACE temp_data
- Although the user music is created, the user cannot start a SQLPlus session because they lack their CREATE SESSION privilege to login to the database.
- A privilege is given to a user with GRANT command
- A role is a collection of privileges and a user granted a role acquires all the privileges defined within the role.
- The following grants some system privileges defined in the Oracle roles CONNECT and RESOURCE
GRANT CONNECT,RESOURCE to music
- Another oracle role is DBA for all system privileges
- You can grant users the privileges to extend privileges they have to other WITH ADMIN option as in:
GRANT CONNECT TO music WITH ADMIN OPTION
- Privileges can be taken away with the REVOKE command as in:
REVOKE RESOURCE FROM music
- A user's password can be changed as:
ALTER USER music IDENTIFIED BY tone DEFAULT TABLESPACE users
- A user is dropped with the following command
DROP USER music CASCADE
- Object privileges can be granted on objects as tables, view, sequences to other users as in :
GRANT SELECT ON course TO music
GRANT ALL ON course TO music WITH GRANT OPTION
- With the second command music is the grantee but can become a granter if the privilege is passed on to another user
- Objects privileges can be revoked as in :
REVOKE UPDATE ON course FROM music

- If you want every user in the database system to have SELECT privileges, you can grant the SELECT privilege to PUBLIC as in:
GRANT SELECT ON instrument to PUBLIC
- A user can define and create ROLE as:
CREATE ROLE read_data_only
SELECT 'GRANT SELECT ON' || table_name || 'to read_data_only;' from user_tables
GRANT read_data_only TO PUBLIC
- To revoke privileges defined by this user defined role, use
REVOKE read_data_only from PUBLIC
- The ability to create roles is only allowed by users with DBA privileges or those granted the CREATE ROEL privilege as in
GRANT CREATE ROLE TO student