April 2017

Practise:-

Creating Table

24 Monday

[114 - 251] Wk 17

Column Name	Datatype	Size	Constraint	Constraint name
user_id	Number	11	Primarykey	PK_USERS
Name	Varchar2	20		
Address	Varchar2	100		
Phno	Number	11		
Emailid	Varchar2	30		

```
create table users(
        user_id number(11),
        name varchar2(20),
        address varchar2(100),
        phno number(11)
        emaiD varchar2(30),
        constraint pk_users primary key(user_id)
        );
```

Alter Table

Write a query to add a new column, LiveTracker of data type char(1). This will be used to track the live location of the bus, for the journey.

desc buses;

Name	Null ?	Type
BUS_NO	NOT NULL	NUMBER(11)
BUS_NAME		VARCHAR2(20)
TYPE		VARCHAR2(20)
TOTAL_SEAT		NUMBER(11)
AVAIL_SEAT		NUMBER(11)

```
alter table buses add
livetracter char(1);
```

Tuesday 25

[115 - 250] Wk 17

Q.1. Write a query to create Distributor table w. constraints ~~method~~ mentioned.

Distributor		
Distributor_ID	varchar(10)	PK
Distributor_Name	varchar(20)	
Address	varchar(100)	
Mobilenumber	number(10)	
Email	varchar(30)	

```
create table distributor(
        distributor_id varchar(10),
        distributor_name varchar(20),
        address varchar(100),
        mobilenumber number(10),
        email varchar(30),
        constraint pk primary key(distributor_id)
        );
```

Q.2. W.A.Q. to create Customer_info table w. constraints mentioned.

← Previously, object were created. Now since the objects are created, it should hold the data.

April 2017

Data Manipulation Language

26 Wednesday (116 - 249) Wk 17

Defines the Data of Table.

1. INSERT
2. UPDATE
3. DELETE
4. SELECT

INSERT ROWS TO TABLE

Syntax → INSERT INTO TableName[(column1, column2...)] VALUES (value1, value2..);

Example → INSERT INTO Customer VALUES (1, 'Tom', 9876523190, 'Tom@g.com', 'Mumbai');

Data for DATE, CHAR and VARCHAR types should be always enclosed within single quotes.

Specific insertion → INSERT INTO Customer(id, cname, phoneno, address) VALUES (2, 'Mini', 91283, 'Chennai');

UPDATE ROWS IN TABLE

UPDATE TableName SET ColumnName = value [, ColumnName = value, ...] [WHERE condition];

April / May

Freedom Day (South Africa)

Thursday 27 (117 - 248) Wk 17

Modify existing rows with the UPDATE statement

```
UPDATE Customer SET
    emailid = 'Tiny15@gmail.com' WHERE
    cname = 'Tiny';
```

All rows in the table are modified if we omit the WHERE clause.

```
UPDATE Customer SET
    emailid = 'Tiny15@gmail.com';
```

DELETE ROWS FROM TABLE

DELETE [FROM] table [WHERE condition];

Remove existing rows from a table by using the DELETE statement.

- DELETE FROM Customer WHERE Cid=2;

To delete all rows from the table.

- DELETE FROM Customer;

- Deletion is not possible if the rows to be deleted is referred in the child table.
- Deletion of the parent record is made possible by using a foreign key reference option.

28 Friday (118 - 247) Wk 17

- CASCADE → Deletes the row from the parent table, & automatically deletes the matching rows in the child table.
- SETNULL → Deletes the row from the parent table, & sets the foreign key column in the child table to NULL.
- RESTRICT → Rejects the delete or update operation for the parent table. This is default option.

Syntax ⇒ REFERENCES tbl_name (index_col_name,...) [ON DELETE reference_option]

- reference_option : RESTRICT | CASCADE | SET NULL

ON DELETE CASCADE

When a record is removed from the customer table, it should also be removed from the related records in the enrollment table.

Use "ON DELETE CASCADE" while creating a table.

```
CREATE TABLE PolicyEnrollment(Enrollment number (5)
primary key, Cid number(10) references
customer (Cid) ON DELETE CASCADE,
Pid varchar2(10), Duedate date,
Paiddate date, penalty number(10));
```


ON DELETE SET NULL

Saturday 29 (119 - 246) Wk 17

If you want to set null values instead of removing the record from the transaction table, how do you do that?

Use "ON DELETE SET NULL' while creating the table.

```
CREATE TABLE PolicyEnrollment (EnrollmentId
number(5) primary key, Cid number(10)
reference customer (Cid) ON DELETE SET NULL,
Pid varchar2(10), Duedate date,
Paiddate date, penalty number(10));
```

DELETE VS TRUNCATE

DELETE	TRUNCATE
- Deletes all rows or a specific row from the table	Removes all rows from the table.
- Can be reverted to its previous state.	Cannot be reverted.
- Rows that are referred in the child table cannot be removed.	Will not work if the truncated table is referenced.

30 Sunday (120 - 245) Wk 18

MERGE:- Provides the ability to conditionally update or insert data into a DB table.

Performs an UPDATE if the row exists & an INSERT if it is a new row.

↳ Avoids separate updates
↳ Increases performance & ease of use.

The MERGE statement is deterministic.

↳ Cannot update the same row of the target table multiple times in the same MERGE statement.

Syntax ⇒

```
MERGE INTO table_name table_alias
    USING (table|view|sub_query) alias
    ON (join condition)
WHEN MATCHED THEN
    UPDATE SET
        col1 = col_val1,
        col2 = col_val
WHEN NOT MATCHED THEN
    INSERT (column_list)
        VALUES (column_values);
```

situation to use MERGE ⇓

Eg: Maintain a table policy history which will allow have the Policy Name & the no. of times the policy has been borrowed.
- Insertion should happen when a policy is taken for the first time & updating of the count for the next time.

Example ⇒

```
MERGE INTO policyhistory as ph
    USING Policy p
    ON (ph.Pname = p.Pname)
WHEN MATCHED THEN
    UPDATE SET ph.count = ph.count + 1

WHEN NOT MATCHED THEN
    INSERT (ph.pname, ph.count)
        VALUES (p.pname, 1);
```

ROLLBACK → Reverting the data.

↳ Reverting the data until a check point is done by save point.

DATABASE TRANSATIONS:-

1. A transaction is a logical unit of work.
2. Oracle ensure data consistency based on transactions.
3. Transaction does consistent data change.
4. Transaction begins when DML statement is executed.
5. Transaction should end with either commit or Rollback.
6. **DDL commands are by default auto commit.**

Commit ⇒ Ends the current transaction by making all pending data changes permanent

SAVEPOINT savepoint_name → Marks a save point within the current transaction.

ROLLBACK → ROLLBACK ends the current transaction by discarding all pending data changes.

Monthly Planner

May 2017

1 Mon	17 Wed
2 Tue	18 Thu
3 Wed	19 Fri
4 Thu	20 Sat
5 Fri	21 Sun
6 Sat	22 Mon
7 Sun	23 Tue
8 Mon	24 Wed
9 Tue	25 Thu
10 Wed	26 Fri
11 Thu	27 Sat
12 Fri	28 Sun
13 Sat	29 Mon
14 Sun	30 Tue
15 Mon	31 Wed
16 Tue	

May Day / Labour / Workers' Day

Monday 1

[121-244] Wk 18

ROLLBACK TO Savepoint_name → ROLLBACK TO SAVEPOINT rolls back the current transaction to the specified savepoint, thereby discarding any changes or savepoints created after the savepoint to which you are rolling back.

LOCKING :- Prevent destructive interaction between concurrent transactions when accessing shared resources.

Locking is performed automatically & requires no user action.

Uses the lowest level of restrictiveness.

Locks are held for a duration of a transaction.

Types -
- Explicit locking
- Implicit locking

May 2017

Tuesday 2

Ques. Insert the following records to the department table.

Department_id	Department_name	Department_block no.
1	Computer Science	3
2	Information Tech.	3
3	Software Engineering	3

Soln. Commands for inserting records to already existing tables is INSERT.

- All the columns have values, so its not necessary for you to bring in column names & then pass on the values.

```
insert into department values (1, 'Computer Science', 3);
insert into department values (2, 'Information Technology', 3);
insert into department values (3, 'Software Engineering', 3);
;
```

* Column names & Table names can be case insensitive but values can't.

Ques. Write a query to change the bus type to 'AC' for all buses.

Buses.

BUS_NO	number(11)
BUS_NAME	varchar2(20)
TYPE	varchar2(20)
TOTAL_SEATS	number(11)
AVAIL_SEATS	number(11)

Soln.

```
update buses set type = 'AC';
```

Select Statement

Wednesday 3

- Also a form of DML statement.
- but mainly used for data retrival statement
- The SELECT statement of SQL is used to retrieve data from one or more tables or views.
- The result returned is stored in a temporary result table, called the result-set.

SELECT can be used for retriving :-

1. all rows from a table
2. specific rows from a table
3. specific values from a table.

SELECT syntax

```
SELECT [distinct][column_name1, column_name2..]*
FROM table_name[alias][, table_name[alias]...
[WHERE conditions]
[GROUP BY group[HAVING group_condition]]
[ORDER BY sort_columns[Asc|Desc]];
```

Note :- SQL statements are not case sensitive, but the data stored in the DB is case sensitive.
Eg. a member with the name TOM & tom are diff., but SELECT & select are the same statement.

May 2017

4 Thursday (124 - 241) Wk 18

Retrive all the records.

Query:

```
SELECT * FROM Customer;
```

(Customer table)

Retrieve Specific Columns:-

Query:

```
SELECT cname, address FROM Customer;
```

CNAME	ADDRESS
Tom	chennai
John	delhi
Ravi	pune
Tiny	chennai

Arithmetic Expressions

- The given query help us to know the min. amount to be paid for the policy including its GST of 10%

PID	PNAME	PPERIODIN YEARS	MINAMOUNTPER MONTH
MBP	Money Back Plan	20	1000
PP	Personal Protect	15	1500

```
SELECT pid, Minamountpermonth +(Minamountpermonth*0.1)
FROM policy;
```

PID	MINAMOUNTPERMONTH +(MINAMOUNTPERMONTH*0.1)
MBP	1100
PP	1650

May

June

Friday 5 (125 - 240) Wk 18

Column Alias

Previous results set has the o/p expression & the entire expression became the column heading bez no column alias was given.

- Here, the result set will have the o/p of the expression.
- Column name is overridden with the alias name provided in the statement.

```
SELECT pid, Minamountpermoth +(Minamount permonth*0.1)
Policy Amount FROM policy;
```

PID	POLICYAMOUNT
MBP	1100
PP	1650

Arithmetic expressions containing a NULL value evaluated to NULL.

- Enclose the alias name in double quotes, if it contains space between words.

Concatenation Operator:-

Ques. How to display the customer name along with their address in the following format "CustomerName" lives in "Address"?

|| operator is used to concatenate columns or character strings.

May 2017

6 Saturday

CUSTOMER ADDRESS
Tom lives in Chennai
John lives in Delhi
Ram lives in Pune
Tiny lives in Chennai

```
SELECT Cname || ' lives in ' || Address as
CustomerAddress FROM customer;
```

Eliminating Duplicate Rows :-

To eliminate duplicate records while retriving we use DISTINCT keyword in SELEC clause.

```
SELECT DISTINCT address FROM customer;
```

ADDRESS
Chennai
Delhi
Pune

WHERE CLAUSE :- The rows return from the query can be restricted by using the WHERE clause.

- A WHERE clause contains the condition that must be met. It directly follows the 'FROM' clause.

```
SELECT *|{[DISTINCT] column|expression[alias],...}
    FROM table[WHERE condition(s)];
```

- Values in the condition is compared using Relational operators.
- Logical operators are used to specify more than one condition in WHERE clause.

Sunday 7

```
SELECT cname, emailid, address FROM customer
    WHERE address = 'Chennai';
```

Comparison Operators :-

=, >, >=, <, <=, (!=, <>)

Between ... and → Range of values.
In (set) → Match any of the list values.
Like → Match a character pattern
Isnull → Is a null value!

```
SELECT * FROM Policy WHERE MinAmountPerMonth
```
< less than operator
```
< 1500;
```

```
SELECT cid, cname, emailid FROM customer
    WHERE emailid LIKE '%yahoo%';
```

Like operator ⇒ '%' denotes zero or many characters
⇒ '_' denotes one character.

Note :- A null is not the same as zero or a space. Zero is a number & space is a character. If any operation has NULL value, then the result is NULL.

May 2017

8 Monday

(128 - 237) Wk 19

OPERATORS	MEANING	EXAMPLE
Between X and Y	Range of values between X and Y (X, Y is inclusive)	.... WHERE Salary Between 5000 and 8000.
In (set)	Match any of the list values	.... WHERE Salary IN (5000, 6000, 7000)
IS NULL	Is a null value	... WHERE Salary IS NULL

- To retrieve policy details where minimum amount to be paid is between 2500 & 5000.

```
SELECT * FROM policy WHERE MinAmountPerMonth
    BETWEEN 2500 AND 5000;
```

- To retrieve customer id & policy id of customers whose customer id is 101, 105

```
SELECT cid, pid FROM policyenrollment WHERE cid in
                                        (101, 105);
```

- To retrieve policy enrollment details of those who have not paid penalty for late payment.

```
SELECT * FROM policyenrollment WHERE penalty
                                    IS NULL;
```

Logical Operators : AND, OR, NOT

⇒ SELECT cid, cname FROM customer WHERE cid = 1 AND address = 'chennai';

May

June

Order by clause

Tuesday 9

(129 - 236) Wk 19

To retrieve names of the customers in ascending order.

```
SELECT cname FROM customer ORDER BY
                                cname;
```

CNAME
John
Mini
Ram
Tiny
Tom

Default :- ascending order

for descending order :-

```
SELECT cname from
customer order by cname desc;
```

Ques. Write a query to display all staff details & sort the result by ID in ascending order.

Staff

staff_id number → PK
staff_name varchar(30)
department_id number → FK

same → select * from staff order by staff_id asc;

Other Example :-

⇒ select * from staff where staff_id > 12 order by staff_id desc;

⇒ select staff_id, staff_name, department_id from staff order by staff_id;

(by - default) ← optional