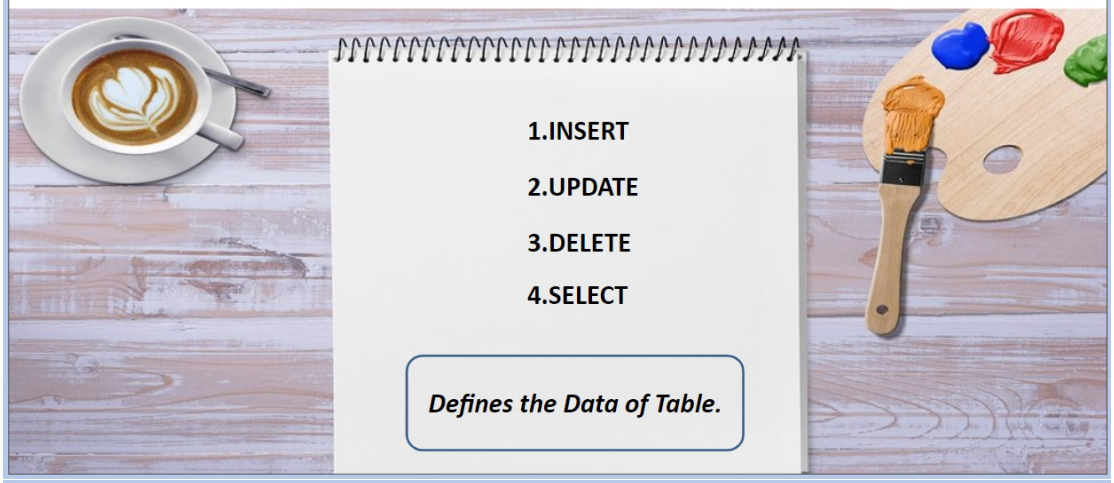
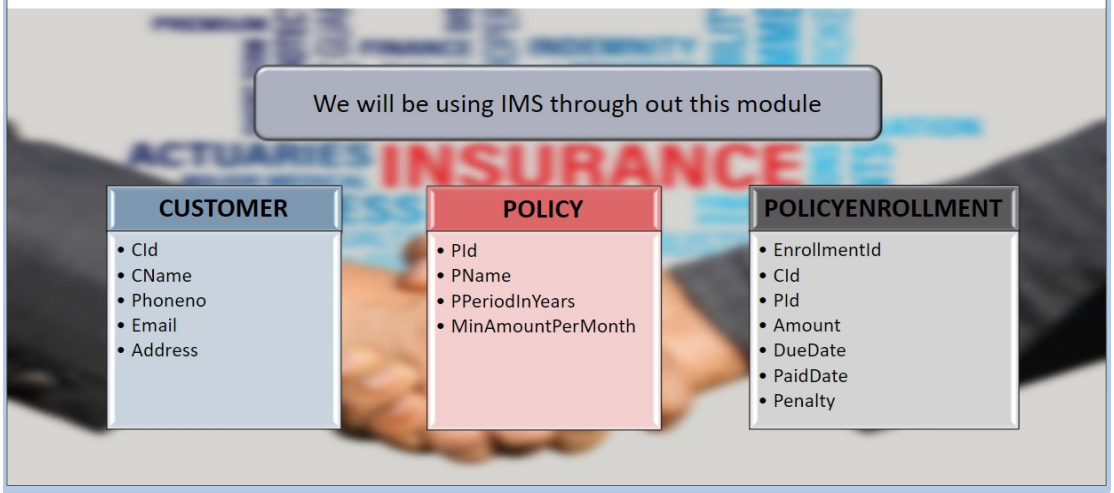
DATA MANIPULATION LANGUAGE



INSURANCE MANAGEMENT SYSTEM(IMS)



SAMPLE RECORDS

CID	CNAME	PHONENO	DOB	EMAILID	ADDRESS
1	Tom	9876523190	17-MAR-87	tom@gmail.com	chennai
2	John	8765432190	26-JAN-86	john@yahoo.com	delhi
3	Ram	7654321890	14-DEC-85	ram@gmail.com	pune
4	Tiny	9012365478	28-MAY-86	NULL	chennai

Customer

PID	PNAME	PPERIODINYEARS	MINAMOUNTPERMONTH
MBP	Money Back Plan	20	1000
PP	Personal Protect	15	1500

Policy

ENROLLMENTID	CID	PID	DUEDATE	PAIDDATE	AMOUNT	PENALTY
101	3	MBP	12-Dec-2017	11-Dec-2017	2000	0
102	1	PP	15-Mar-2018	13-Mar-2018	3000	0
103	2	PP	15-Feb-2018	22-Feb-2018	4000	200

Policy Enrollment

INSERT ROWS TO TABLE

Syntax:

INSERT INTO *TableName [(column 1, column2...)]* **VALUES** *(value1 , value2...)*;

Example:
INSERT INTO Customer **VALUES** (1,'Tom',9876523190,'Tom@gmail.com','Mumbai');

INSERT ROWS TO TABLE

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',9128374605,'Chennai');

UPDATE ROWS IN A TABLE

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

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

TEKNOTURF

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 row to be deleted is referred in the child table.

Deletion of the parent record is made possible by using a foreign key reference option.

DELETE ROWS FROM TABLE

TEKNOTURF

REFERENCE OPTION

- **CASCADE** : Deletes the row from the parent table, and automatically deletes the matching rows in the child table.
- **SET NULL** : Deletes the row from the parent table, and 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

TEKNOTURF

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(EnrollmentId 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

TEKNOTURF

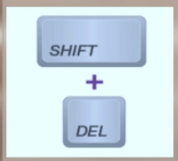
If you want to set null values instead of removing the record from the transaction table, how do you do that?

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

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

DELETE VS TRUNCATE

TEKNOTURF

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

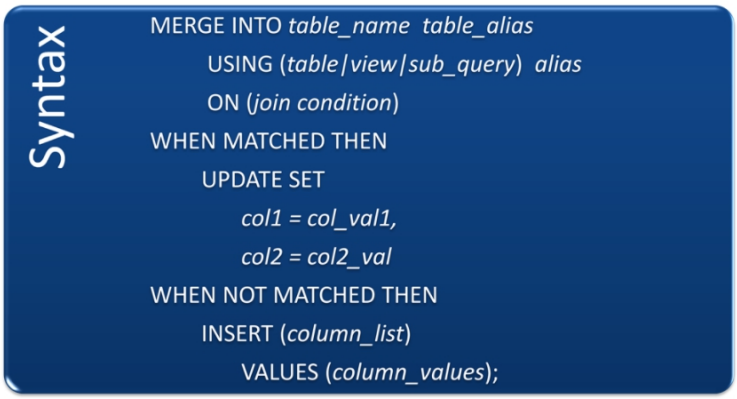
MERGE

TEKNOTURF

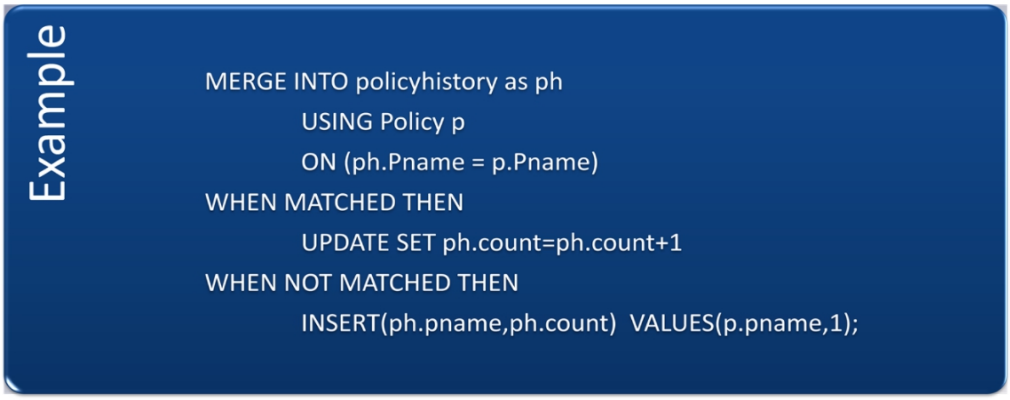
Provides the ability to conditionally update or insert data into a database table.

Performs an UPDATE if the row exists and an INSERT if it is a new row.	• Avoids separate updates. • Increases performance and ease of use.
The MERGE statement is deterministic .	• Cannot update the same row of the target table multiple times in the same MERGE Statement.

MERGE



MERGE



DATABASE TRANSACTION

- A transaction is a logical unit of work.
- Oracle ensures data consistency based on transactions.
- Transaction does consistent data change.
- Transaction begins when DML statement is executed.
- Transaction should end with either **Commit or Rollback.**
- DDL commands are by default auto commit.

DATABASE TRANSACTION

TEKNOTURF

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.
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

TEKNOTURF

Prevent destructive interaction between concurrent transactions when accessing shared resources.

Locking is performed automatically and requires no user action.

Uses the lowest level of restrictiveness.

Locks are held for a duration of a transaction.

Types

- Explicit locking
- Implicit locking

Insert Records Demo

Description Edit Grading view

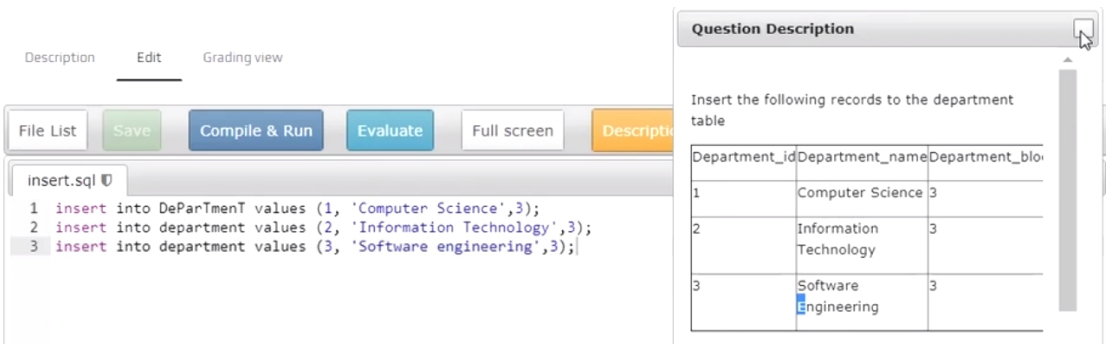
Insert Records - Department

Requested files: insert.sql
(Download)

Type of work: Individual work
Contains the run and evaluate script of Oracle DML. Can be used as example

Insert the following records to the department table

Department_id	Department_name	Department_block_number
1	Computer Science	3
2	Information Technology	3
3	Software Engineering	3



Update Records Demo

Description Edit Grading view

Update Buses table

Requested files: update1.sql
(Download)

Type of work: Individual work
Contains the run and evaluate script of Oracle DML. Can be used as example

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