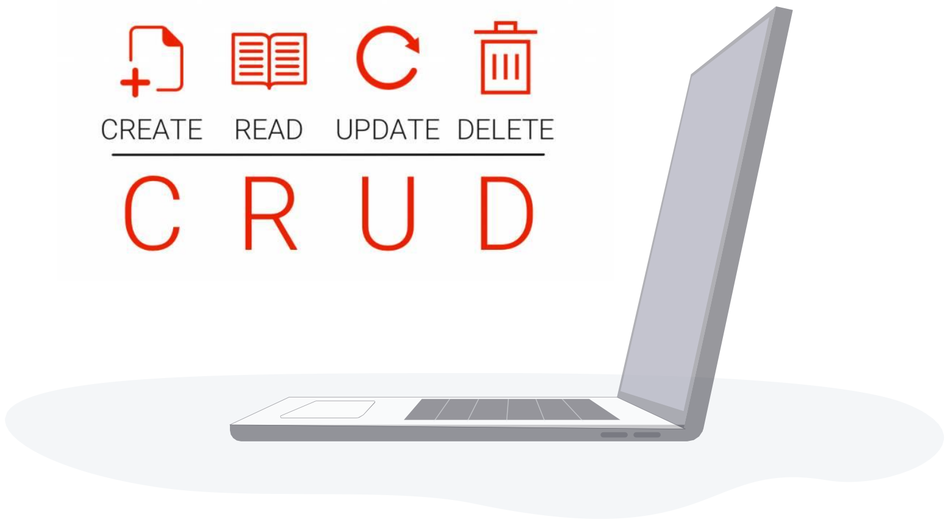
Harmony IT Solution

Database Design and Programming

Tahaluf Training Center 2022

Harmony IT Solution

1. Overview of Data Manipulation Language (DML)
2. Data Manipulation Language (DML) Commandes



Harmony IT Solution

Overview of Data Manipulation Language (DML)

- **Data Manipulation Language (DML)** is a language used to select, insert, update and delete data from a database.

- DML used to manipulate and retrieve data in a relational database.

Harmony IT Solution

DML commands are as follows:

- INSERT.
- UPDATE.
- DELETE.
- SELECT.

Note: DML performs read-only queries of data.

Exercise

Create a database for the following scenario:

In an exam system, each course contains a many of exams. Each course has a name, id, description, start date and end date. Each exam has name, id, duration, start time.

Harmony IT Solution

Exercise Solution

```
CREATE TABLE Course(
    Id NUMBER GENERATED BY DEFAULT AS IDENTITY,
    Name VARCHAR2(50) NOT NULL,
    Description VARCHAR2(250) NOT NULL,
    StartDate DATE NOT NULL,
    EndDate DATE NOT NULL,
    PRIMARY KEY(Id)
);
```

```
CREATE TABLE Exam (
 Id NUMBER GENERATED BY DEFAULT AS IDENTITY,
 Name VARCHAR2(50) NOT NULL,
 Duration TIMESTAMP NOT NULL,
 StartTime TIMESTAMP NOT NULL,
 courseID NUMBER,
 CONSTRAINT FKeyCourse_Exam
 FOREIGN KEY (courseID)
 REFERENCES Course (Id)
);
```

Harmony IT Solution

INSERT command

- **INSERT command** is used to insert a data into a table.

- This command can used to add one or more records to a single table in a database.

```
INSERT INTO table(column, column...)
VALUES (value, value...);

INSERT INTO table
VALUES (value, value...);
```

There are many mistakes that users make with the insert statement:

- ✓Duplicate value violating any primary or unique key constraint.
- ✓Forgetting a mandatory value for a NOT NULL column.
- ✓Referential integrity maintained for foreign key constraint.
- ✓Any value violating a CHECK constraint.
- ✓Values too **wide** to **fit** in column or **data type mismatches**.

Example

Insert a record into course table.

Example Solution

```
INSERT INTO Course(Name, Description, StartDate,EndDate)
VALUES ('Math101', 'Maths', DATE'2021-01-01', DATE'2021-10-01');
```

Exercise:

Insert a three record into Course table. Then, Insert a five record into Exam table.

Harmony IT Solution

Exercise Solution

```sql
INSERT INTO Course(Name, Description, StartDate,EndDate)
VALUES ('IT101', 'IT', DATE'2021-02-15', DATE'2023-08-01');

INSERT INTO Course(Name, Description, StartDate,EndDate)
VALUES ('English101', 'English', DATE'2050-02-09', DATE'2070-08-01');

INSERT INTO Course(Name, Description, StartDate,EndDate)
VALUES ('English102', 'English', DATE'2000-02-09', DATE'2010-12-05');

INSERT INTO Exam(Name, Duration, StartTime,Id)
VALUES ('FirstExam',timestamp '2017-10-12 21:22:23', CURRENT_TIMESTAMP, 2);

INSERT INTO Exam(Name, Duration, StartTime,Id)
VALUES ('FirstExam',timestamp '2021-09-16 01:30:23', timestamp '2021-09-16 3:22:23', 2);

INSERT INTO Exam(Name, Duration, StartTime,Id)
VALUES ('SecondExam',timestamp '2021-09-16 01:30:23', timestamp '2021-09-16 3:22:23', 4);
```

Harmony IT Solution

UPDATE command

- **UPDATE command** is used to modify the records in a table.

- This command is used to update existing data changes it of one or more records in a table.

```
UPDATE table
SET column = value [, column = value, ...]
[WHERE condition];
```

Example

Update a course name if the description contains English.

Example Solution

```
UPDATE Course
SET Name = 'English100'
WHERE Description = 'English';
```

Harmony IT Solution

DELETE command

- **DELETE command** is used to delete or remove a single record or multiple records from a table.

```
DELETE FROM table WHERE condition;
```

Example

Delete a course if the course name is IT 101.

Example Solution

```
DELETE FROM Course
WHERE Name = 'IT101';
```

Harmony IT Solution

TRUNCATE TABLE

Harmony IT Solution

- **TRUNCATE TABLE** command is used to delete all records from a table.
- It is the same command as a DELETE without a WHERE clause.

```
TRUNCATE TABLE table_name;
```

If you need to truncate a table, the TRUNCATE TABLE command can not be rolled back.

Example

Delete all record from course table.

Example Solution

```
TRUNCATE TABLE Course;
```

Harmony IT Solution

SELECT statement

A SELECT statement retrieves data from database. It is used to:

1. **Projection:** select the columns in a table that are returned using a query.

2. **Selection:** select the rows in a table that are returned using a query.

3. **Joining:** bring together data that is stored in different tables.

SELECT Example:

```
SELECT id, name, description, startdate, enddate
FROM Course;
```

	ID	NAME	DESCRIPTION	STARTDATE	ENDDATE
1	15	English101	English	09-FEB-50	01-AUG-70
2	16	English102	English	09-FEB-00	05-DEC-10
3	17	IT101	IT	15-FEB-21	01-AUG-23
4	18	English101	English	09-FEB-50	01-AUG-70
5	19	English102	English	09-FEB-00	05-DEC-10

SELECT Example:

```
SELECT * FROM Course;
```

	ID	NAME	DESCRIPTION	STARTDATE	ENDDATE
1	15	English101	English	09-FEB-50	01-AUG-70
2	16	English102	English	09-FEB-00	05-DEC-10
3	17	IT101	IT	15-FEB-21	01-AUG-23
4	18	English101	English	09-FEB-50	01-AUG-70
5	19	English102	English	09-FEB-00	05-DEC-10

Harmony IT Solution

Arithmetic Operators

- **Arithmetic Operators** are used to add, negate, multiply, subtract, and divide **numeric** values:

  1. **(+ , -)** : Unary operators denotes a negative or positive expression.

  2. **(*, /)** : Binary operators represent a multiplies and divides.

  3. **(+, -)** : Binary operators represent adds, subtracts.

Harmony IT Solution

Arithmetic Operators Examples:

```
SELECT id+5, name, description, startdate, enddate
FROM Course;
```

	ID+5	NAME	DESCRIPTION	STARTDATE	ENDDATE
1	20	English101	English	09-FEB-50	01-AUG-70
2	21	English102	English	09-FEB-00	05-DEC-10
3	22	IT101	IT	15-FEB-21	01-AUG-23
4	23	English101	English	09-FEB-50	01-AUG-70
5	24	English102	English	09-FEB-00	05-DEC-10

Arithmetic Operators Examples:

```
SELECT name, 5+mark*12, id
FROM Student;
```

	NAME	5+MARK*12	ID
1	ahmad	1085	1
2	ali	965	2
3	sami	1061	3
4	fade	1025	4

Harmony IT Solution

Arithmetic Operators Examples:

```
SELECT name, (5+mark)*12, id
FROM Student;
```

	NAME	(5+MARK)*12	ID
1	ahmad	1140	1
2	ali	1020	2
3	sami	1116	3
4	fade	1080	4

ALIASES Statement

- **ALIASES Statement** is used to create a temporary name for tables or columns.

- COLUMN ALIASES are used to make column names easier to read.

```
column_name AS alias_name
```

Harmony IT Solution

COLUMN ALIASES Example

```
SELECT name, (5+mark)*12 AS IncreaseSalary, id FROM Student;
```

	NAME	INCREASESALARY	ID
1	ahmad	1140	1
2	ali	1020	2
3	sami	1116	3
4	fade	1080	4
5	H	(null)	1

TABLE ALIASES are used to make a table easier to read or to perform a self join.

```
table_name alias_name
```

TABLE ALIASES Example :

```
SELECT name, (5+mark)*12, id FROM Student Std;
```

	NAME	(5+MARK)*12	ID
1	ahmad	1140	1
2	ali	1020	2
3	sami	1116	3
4	fade	1080	4
5	H	(null)	1

DISTINCT Operator :

exclude duplicate rows.

Example

```
SELECT DISTINCT name, startdate, enddate, description FROM
Course;
```

	NAME	STARTDATE	ENDDATE	DESCRIPTION
1	English102	09-FEB-00	05-DEC-10	English
2	English101	09-FEB-50	01-AUG-70	English
3	IT101	15-FEB-21	01-AUG-23	IT

WHERE Clause :

is used to restrict the rows returned.

Example

```
SELECT * FROM Student WHERE MARK > 80;
```

	NAME	MARK	ID
1	ahmad	90	1
2	sami	88	3
3	fade	85	4

Equal Operator

```
SELECT * FROM Student WHERE MARK = 90;
```

	NAME	MARK	ID
1	ahmad	90	1

Not equal Operator

```
SELECT * FROM Student WHERE MARK <> 90;
```

OR

```
SELECT * FROM Student WHERE MARK != 90;
```

	NAME	MARK	ID
1	ali	80	2
2	sami	88	3
3	fade	85	4

Harmony IT Solution

Greater Than or Equal Operator

```
SELECT * FROM Student WHERE MARK >= 85;
```

	NAME	MARK	ID
1	ahmad	90	1
2	sami	88	3
3	fade	85	4

Less Than Operator

```
SELECT * FROM Student WHERE MARK < 85;
```

	NAME	MARK	ID
1	ali	80	2

Less Than or Equal Operator

```
SELECT * FROM Student WHERE MARK <= 85;
```

	NAME	MARK	ID
1	ali	80	2
2	fade	85	4

IN Operator : Matches a value in a list.

```
SELECT * FROM Student WHERE MARK IN (90, 80);
```

	NAME	MARK	ID
1	ahmad	90	1
2	ali	80	2

NOT Operator : Negates a condition.

```
SELECT * FROM Student WHERE MARK NOT IN (90, 80);
```

	NAME	MARK	ID
1	sami	88	3
2	fade	85	4

BETWEEN Operator : Within a range (inclusive).

```
SELECT * FROM Student WHERE MARK BETWEEN 85 AND 90;
```

	NAME	MARK	ID
1	ahmad	90	1
2	sami	88	3
3	fade	85	4

Harmony IT Solution

IS NULL Operator : NULL value.

```
SELECT * FROM Student WHERE MARK IS NULL;
```

	NAME	MARK	ID
1	H	(null)	1

Harmony IT Solution

IS NOT NULL Operator : Non-NULL value.

```
SELECT * FROM Student WHERE MARK IS NOT NULL;
```

	NAME	MARK	ID
1	ahmad	90	1
2	ali	80	2
3	sami	88	3
4	fade	85	4

Harmony IT Solution

LIKE Examples

```
SELECT * FROM Student WHERE Name LIKE '%i';
```

	NAME	MARK	ID
1	ali	80	2
2	sami	88	3

LIKE Examples

```
SELECT * FROM Student WHERE Mark LIKE '8_';
```

	NAME	MARK	ID
1	ali	80	2
2	sami	88	3
3	fade	85	4

Logical Conditions combines the results of two conditions to produce a single result.

1. **NOT** returns TRUE if the condition is FALSE.

2. **AND** returns TRUE if both conditions are TRUE.

3. **OR** returns TRUE if either condition is TRUE.

Harmony IT Solution

Logical Conditions Examples

```
SELECT * FROM Student WHERE name LIKE 'a%' OR mark = 90;
```

	NAME	MARK	ID
1	ahmad	90	1
2	ali	80	2

Harmony IT Solution

Logical Conditions Examples

```
SELECT * FROM Student WHERE name LIKE 'a%' And mark = 90;
```

	NAME	MARK	ID
1	ahmad	90	1

Harmony IT Solution

Logical Conditions Examples

```
SELECT * FROM Student WHERE name NOT LIKE'%i';
```

	NAME	MARK	ID
1	ahmad	90	1
2	fade	85	4
3	H	(null)	1

Harmony IT Solution

ORDER BY Clause

- In Oracle, **ORDER BY** Clause is used to sort or re-arrange the records in the result set.

- The **ORDER BY** clause is only used with SELECT statement.

```
SELECT expressions
FROM tables
WHERE conditions
ORDER BY expression [ ASC | DESC ];
```

Example:

```
SELECT * FROM Student WHERE name NOT LIKE'%i’ ORDER BY ID
ASC;
```

Note: ORDER BY ASC is the default case.

	NAME	MARK	ID
1	ahmad	90	1
2	H	(null)	1
3	fade	85	4

Example:

```
SELECT * FROM Student WHERE name NOT LIKE'%i’ ORDER BY ID
DESC;
```

	NAME	MARK	ID
1	fade	85	4
2	H	(null)	1
3	ahmad	90	1

Harmony IT Solution

SQL function

Harmony IT Solution

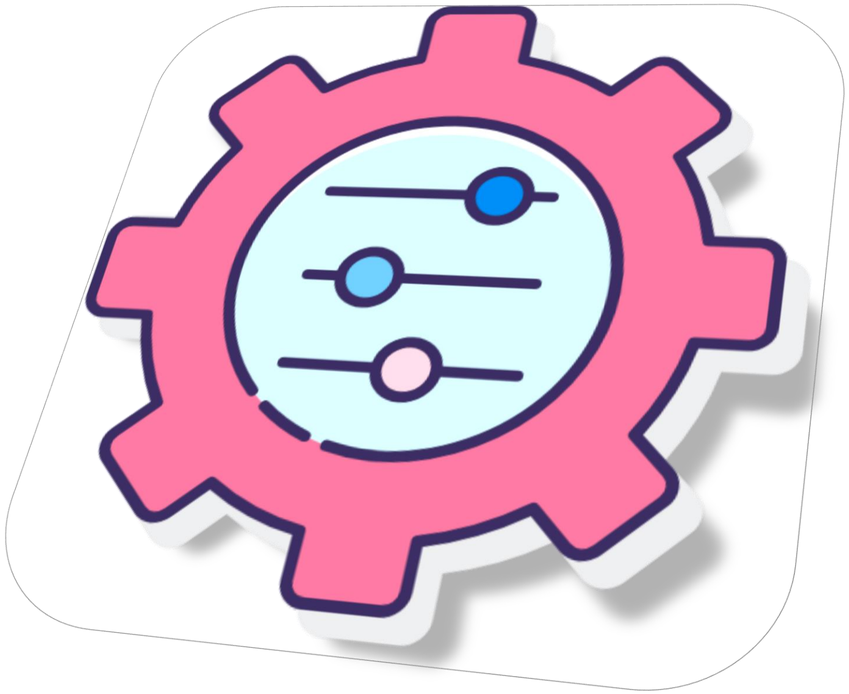
SQL function :

1. Single Row Function:

a) Character.
b) Number.
c) Date.
d) Conversion.

2. Multiple Row Function:

a) Aggregate function GROUP BY.



Character functions example:

```
1. SELECT INITCAP(name) from Student;

2. SELECT LENGTH(name) from Student;

3. SELECT ROUND(mark) from Student;

4. SELECT TRUNC(mark) from Student;
```

Harmony IT Solution

General function example:

1. SELECT MOD(mark, id) from Student;

2. SELECT NVL(name, 0) FROM employee;

	NAME
1	khaled
2	sami
3	vaseen
4	(null)
5	alaa
6	(null)

SELECT NVL(name, 0) FROM employee;

NVL(NAME,0)
khaled
sami
vaseen
0
alaa
0

Aggregate function GROUP BY example:

```
1. SELECT name, Min(Mark) AS "Min" FROM student GROUP BY name;

2. SELECT name, Max(Mark) AS "Max" FROM student GROUP BY name;

3. SELECT name, COUNT(*) FROM student GROUP BY name;

4. SELECT name, COUNT(*) FROM student WHERE Id= 1 GROUP BY name;

5. SELECT name FROM student HAVING SUM(mark)>=90 GROUP BY name;
```

Harmony IT Solution

References :

[1]. https://www.javatpoint.com/oracle-delete
[2]. https://www.techonthenet.com/oracle/truncate.php
[3]. https://en.wikibooks.org/wiki/Oracle_Database/SELECT_Statement
[4]. https://ramkedem.com/en/oracle-where-clause/
[5].https://docs.oracle.com/cd/A87860_01/doc/server.817/a85397/operator.htm
[6]. https://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqlj13658.html
[7]. https://www.javatpoint.com/oracle-group-by-clause

Harmony IT Solution