alter session set "_ORACLE_SCRIPT"=true;
username : dBmS!190
connect system@orclpdb

CREATE USER username IDENTIFIED BY password;

GRANT CREATE SESSION, RESOURCE, UNLIMITED TABLESPACE TO username;

Connect username;

CREATE TABLE table_name
(
column_1 INT,
column_2 VARCHAR2(size)
// PRIMARY KEY(column_n)
);

SELECT * FROM table_name;

SELECT column1, column3 FROM table_name;

INSERT INTO table_name VALUES (value1, value2,..);
INSERT INTO table_name (column1, column2, column4, ..) VALUES (value1, value2, value3, ..);
• For string use ‘string’

DROP TABLE table_name;

❒ **PRIMARY KEY**

CREATE TABLE table_name
(
column_1 INT **PRIMARY KEY**,
column_2 VARCHAR2(size) ,
// **PRIMARY KEY** (column_n, column_m)
);

Or**, CONSTRAINT** contraint_table_pk **PRIMARY KEY** (col_name / s)

❒ **FOREIGN KEY**

CONSTRAINT contraint _table_fk **FOREIGN KEY** (col_1) **REFERENCES** other_table_name (col_name_2)

ALTER TABLE table_name **ADD CONSTRAINT** constraint_name CONSTRAINT1,CONSTRAINT2...

ALTER TABLE table_name **ADD** column_name datatype;

ALTER TABLE table_name **DROP COLUMN** column_name;

ALTER TABLE table_name **MODIFY** column_name **datatype**;

ALTER TABLE table_name **DROP CONSTRAINT** constraint_name;

SELECT * **FROM** table1,table2... (CARTESIAN PRODUCT)

SELECT * FROM table_name **WHERE** condition; // WHERE NOT

(>,<,=,>=,<=,!=,BETWEEN val1 AND val2 ,OR,LIKE IN)

SELECT * FROM table_name **WHERE** column_name **IN** (val1 , val2);

{ **SELECT * FROM** Customers **WHERE** Country **IN** (**SELECT** Country **FROM** Suppliers); }

SELECT * FROM table_name **WHERE** column_name **BETWEEN** val1 **AND** val2 ; //NOT BETWEEN

SELECT * col_1,col_2, .. **FROM** table_1 **NATURAL JOIN** table_2 ;

SELECT t1. col_1 , t2. col_2, .. **FROM** table_1 **NATURAL JOIN** table_2 **ON** condition ;

SELECT * **FROM** table_1 , table_2 **WHERE** table1.col2 = table2.col1;

SELECT * FROM table_name **ORDER BY** col1 **ASC**, col 2 **DESC** ; // DEF ASC

UPDATE table_name **SET** col1 = val **WHERE** col2=val ;

DELETE FROM table_name **WHERE** cond;

SELECT DISTINCT col1,col2 , **FROM** table_name;

ALTER TABLE table_name **RENAME COLUMN** old_name **TO**

new_name;

[SELECT name new_name, (salary * 1.5) Enhanced_salary
FROM table ;]

SELECT tb1.col1 , tb2.col2 **FROM** table_1 tb1 , table_2 tb2 **WHERE** tb1.col2 = tb2.col3 **AS** new_name;

SELECT col **AS** alias;

❒ **STRING OPERATIONS**

- Built in functions
i) lower
ii) upper
iii) Concatenation ||
iv) Complete matching =
v) Partial matching LIKE

LIKE Operator

Followed by two special characters

% → any substring (Null acceptable, no or more substring)

_ → one character

Example:

‘ Intro% ’ → return anything starting with intro (intro itself too)

‘ %Data% ‘ → any string having data as substring

‘ _ _ ‘ → return string having exactly 2 characters

‘ _ _ % ‘ → at least 2 characters

SELECT * FROM Customers WHERE ContactName LIKE 'a%o';

NVL (salary,0) * 1.25 ; NULL Salaries will be converted to 0

SELECT AVG (col_name) **FROM** table_name **WHERE** condition **;**

SELECT dept, avg(salary) avg_salary **FROM** emp **GROUP BY** dept**;**

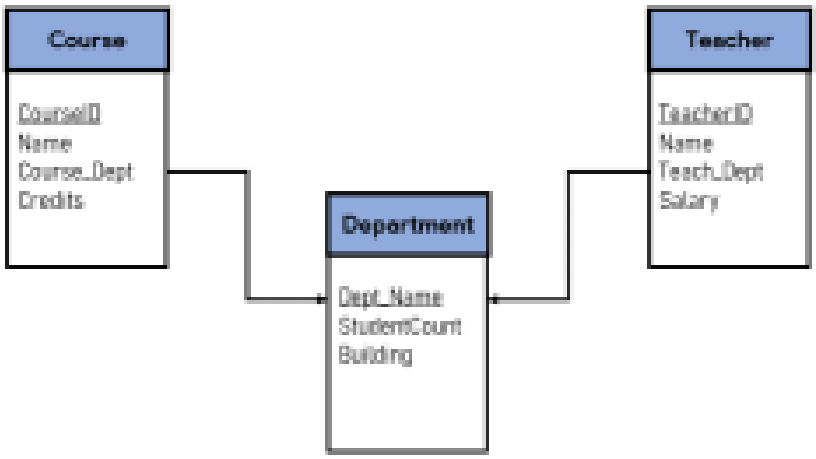
UPDATE Emp SET Salary = Salary * 1.1 WHERE Salary < (SELECT avg (Salary) FROM emp);

COMMIT;

DISCONNECT;

CL SCR;
DESC table_name;

SCHEMA DIAGRAM



→ From **FK** attribute of **referencing relation** to the **PK** of the **referenced relation**

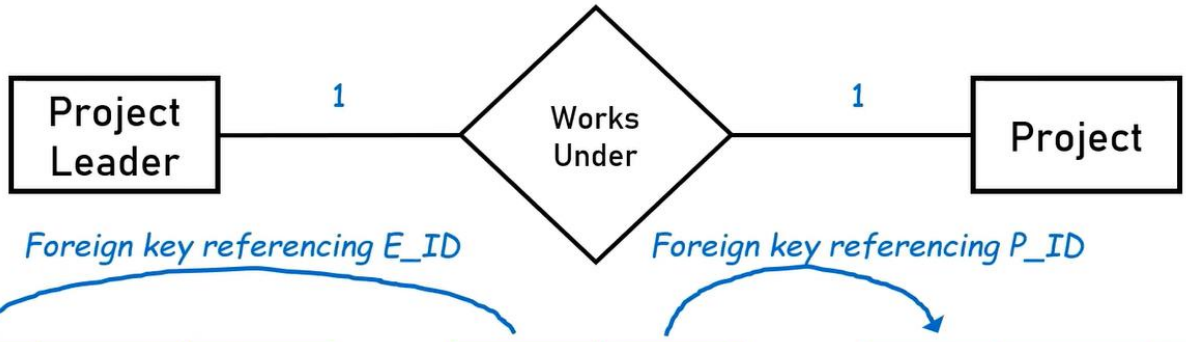
SELECT OCCUPATION, COUNT (OCCUPATION) FROM CITIZEN GROUP BY OCCUPATION;

SELECT C_NAME,C_HOME,C_ID FROM CITIZEN WHERE C_HOME LIKE 'D%';

SELECT OCCUPATION, COUNT (OCCUPATION) FROM CITIZEN GROUP BY OCCUPATION ORDER BY COUNT(OCCUPATION) DESC FETCH FIRST 5 ROWS ONLY;

Converting an ER Model to a Relational Model

Case 1: One-to-One Relationship



Foreign key referencing E_ID

Foreign key referencing P_ID

E_ID	Name	Phone
E1	A	123
E2	B	213
E3	C	321
E4	D	221

E_ID	P_ID
E1	P1
E2	P2
E3	P3

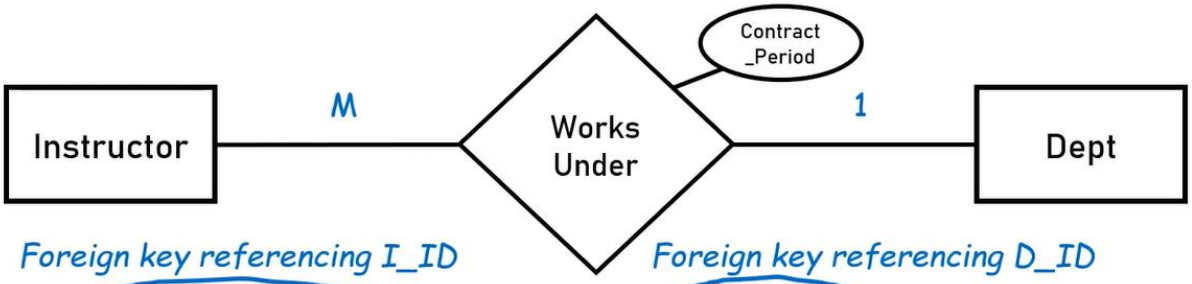
P_ID	Name	Budget
P1	Alpha	100$
P2	Beta	200$
P3	Gamma	1000$

E_ID	Name	Phone	P_ID
E1	A	123	P1
E2	B	213	P2
E3	C	321	P3
E4	D	221	------

P_ID	Name	Budget
P1	Alpha	100$
P2	Beta	200$
P3	Gamma	1000$

Converting an ER Model to a Relational Model

Case 2: Many-to-One Relationship



Foreign key referencing I_ID

Foreign key referencing D_ID

I_ID	Name	Phone
I1	A	123
I2	B	213
I3	C	321
I4	D	221

I_ID	D_ID	Con_ per
I1	D1	1 yr
I2	D1	1 yr
I3	D3	3 yr
I4	D2	2 yr

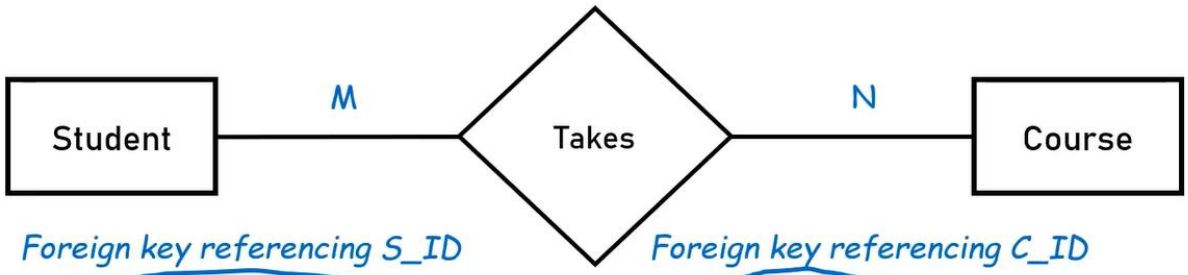
D_ID	Name	Location
D1	CSE	1st flr
D2	EEE	2nd flr
D3	MCE	3rd flr

I_ID	Name	Phone	D_ID	Con_p er
I1	A	123	D1	1 yr
I2	B	213	D1	1 yr
I3	C	321	D3	3 yr
I4	D	221	D2	2 yr

D_ID	Name	Location
D1	CSE	1st flr
D2	EEE	2nd flr
D3	MCE	3rd flr

Converting an ER Model to a Relational Model

Case 3: Many-to-Many Relationship



Foreign key referencing S_ID

Foreign key referencing C_ID

S_ID	Name	Phone
S1	A	123
S2	B	213
S3	C	321
S4	D	221

S_ID	C_ID
S1	C1
S2	C2
S1	C2
S2	C1
S3	C3

C_ID	Name	Credit
C1	DBMS	3.0
C2	C++	3.0
C3	DS	2.0

1 : M – F.K in 'MANY' part
1 : 1 – F.K (in dependent) + UNIQUE
M : M – New Entity (Junction / Association table)
Primary key of both entities will come together
along with some other things