

SELECT \* FROM dept;

SELECT \* FROM project;

SELECT \* FROM employee;

INSERT INTO employee VALUES

(19,1,'sunt','et','Operations Manager',500406,'2020-09-02'),

(54,3,'omnis','quidem','exercitationem',241123,'2013-02-24'),

(78,2,'velit','ab','Operations Analyst',394047,'1985-11-16'),

(80,1,'accusamus','qui','Operations Manager',622295,'2012-08-28'),

(84,1,'et','ad','Auditor',801070,'2020-05-01'),

(121,3,'architecto','qui','Auditor',803625,'2018-08-15'),

(124,3,'dolorum','non','Financial Analyst',913504,'1977-12-26'),

(129,1,'aspernatur','magni','Credit Authorizer',864120,'1985-12-08'),

(132,2,'quia','tempora','Continuous Improvement Lead',751516,'2022-02-18'),

(144,2,'aut','eveniet','Continuous Improvement Consultant',664481,'1986-01-11')

;

INSERT INTO project VALUES

(11,1,'voluptatum','New Karen',2212462,2003),

(17,1,'odio','New Samantha',3031102,1997),

(43,3,'debitis','East Marjoryfort',2621647,2003),

(46,2,'beatae','New Abdiel',4670329,2019),

(47,2,'explicabo','South Gladys',4373152,1991),

(108,3,'sunt','Kreigerville',1252457,1991),

(115,3,'maiores','West Ike',2387308,1983),

(141,1,'quis','Kariside',2332175,2005),

(143,2,'rerum','East Reinhold',1199693,2015),

(150,1,'est','Reingerstad',3076611,1972)

;

SELECT \* FROM employee

WHERE (dept\_id IN (SELECT dept\_id FROM dept WHERE (dept\_name='Computer' OR dept\_name='IT')))

AND (emp\_fname LIKE 'p%' OR emp\_fname LIKE 'h%');

SELECT emp\_position FROM employee GROUP BY emp\_position HAVING COUNT(emp\_position) = 1;

UPDATE employee

SET emp\_salary = emp\_salary\*1.1

WHERE emp\_jdate < '1985-01-01';

DELETE FROM dept WHERE dept\_loc = 'Mumbai';

SELECT proj\_name FROM project WHERE proj\_location = 'Pune';

SELECT \* FROM project WHERE proj\_cost BETWEEN 1000000 AND 5000000;

SELECT proj\_name, MAX(proj\_cost) AS "Maximum Cost", AVG(proj\_cost) AS "Average Project Cost"

FROM project GROUP BY proj\_name;

SELECT emp\_id, emp\_fname, emp\_lname

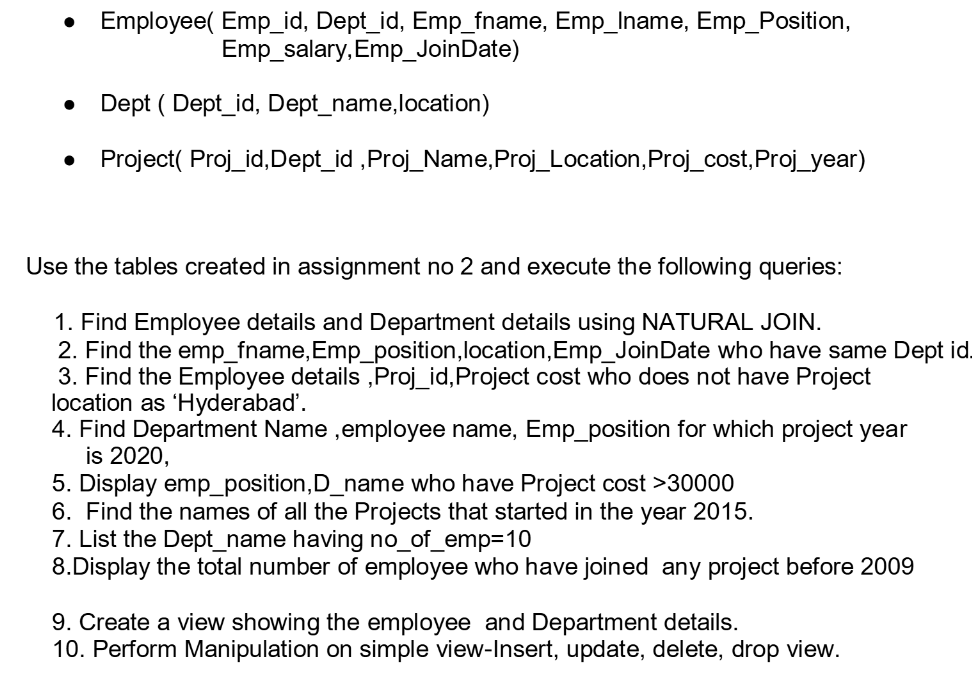
FROM employee

ORDER BY emp\_lname DESC;

SELECT proj\_name AS "Name", proj\_location AS "Location", proj\_cost AS "Cost"

FROM project

WHERE proj\_year = 2004 OR proj\_year = 2005 OR proj\_year = 2007;



USE 31380\_db;

-- Find Employee details and Department details using NATURAL JOIN.

SELECT emp.emp\_id AS "Employee ID",

emp.emp\_fname AS "Name",

emp.emp\_lname AS "Surname",

emp.emp\_position AS "Position",

dep.dept\_name AS "Department",

dep.dept\_loc AS "Dept. Location",

emp.emp\_salary AS "Salary",

emp.emp\_jdate AS "Joining Date"

FROM employee AS emp

NATURAL JOIN dept AS dep

;

-- Find the emp\_fname,Emp\_position,location, Emp\_JoinDate who have same Dept id.

SELECT emp.emp\_fname AS "Name",

emp.emp\_position AS "Position",

dep.dept\_loc AS "Location",

emp.emp\_jdate AS "Joining Date",

dep.dept\_name AS "Department"

FROM employee as emp

LEFT JOIN dept as dep

ON emp.dept\_id = dep.dept\_id

;

-- Find the Employee details ,Proj\_id,Project cost who does not have Project location as ‘Hyderabad’.

SELECT DISTINCT

e.emp\_id,

e.emp\_fname,

e.emp\_lname,

e.emp\_position,

e.emp\_salary,

e.emp\_jdate,

p.proj\_id,

p.proj\_cost

FROM

employee e

INNER JOIN

project p ON e.dept\_id = p.dept\_id

WHERE

p.proj\_location <> 'Hyderabad';

-- Find Department Name ,employee name, Emp\_position for which project year is 2020

SELECT

edj.dept\_name AS "Dept. Name",

edj.emp\_fname AS "Employee Name",

edj.emp\_position AS "Position"

FROM (

SELECT

employee.dept\_id,

employee.emp\_fname,

employee.emp\_position,

dept.dept\_name

FROM employee

LEFT JOIN dept

ON

employee.dept\_id = dept.dept\_id

) AS edj

RIGHT JOIN project AS pr

ON

edj.dept\_id = pr.dept\_id

WHERE

pr.proj\_year = 2020

;

-- OR

SELECT dept\_name, emp\_fname, emp\_position FROM

employee LEFT JOIN dept USING (dept\_id) RIGHT JOIN project USING (dept\_id)

WHERE proj\_year = 2005

ORDER BY emp\_id ASC;

-- Display emp\_position, dept\_name who have Project cost >30000

SELECT

edj.emp\_position AS "Position",

edj.dept\_name AS "Dept. Name"

FROM (

SELECT \*

FROM employee

NATURAL JOIN dept

WHERE employee.dept\_id = dept.dept\_id

) AS edj

RIGHT JOIN project AS pr

ON

edj.dept\_id = pr.dept\_id

WHERE

pr.proj\_cost > 30000

;

-- Find the names of all the Projects that started in the year 2015.

SELECT

pr.proj\_name AS "Project Name"

FROM

project AS pr

WHERE

pr.proj\_year = 2015

;

-- List the Dept\_name having no\_of\_emp = 10

SELECT

edj1.dept\_name

FROM (

SELECT

COUNT(EDJ.emp\_id) AS "CNT", EDJ.dept\_name

FROM (

SELECT

dept.dept\_id,

dept.dept\_name,

employee.emp\_id

FROM dept

LEFT JOIN employee

ON

dept.dept\_id = employee.dept\_id

) as EDJ

GROUP BY EDJ.dept\_id

) AS edj1

WHERE CNT = 10;

-- Display the total number of employee who have joined any project before 2009

-- (Used complex view)

CREATE VIEW edv AS

SELECT d.dept\_id, e.emp\_fname, p.proj\_name, p.proj\_year

FROM employee AS e, dept AS d, project AS p

WHERE d.dept\_id = e.dept\_id = p.dept\_id

ORDER BY dept\_id

;

SELECT DISTINCT emp\_fname

FROM edv

WHERE proj\_year < 2009

;

-- Create a view showing employee and dept details

CREATE VIEW ed\_deets AS

SELECT

e.dept\_id,

e.emp\_fname,

e.emp\_lname,

e.emp\_position,

e.emp\_salary,

e.emp\_jdate,

d.dept\_name,

d.dept\_loc

FROM

employee AS e,

dept AS d

WHERE

e.dept\_id = d.dept\_id

ORDER BY

dept\_id

;

SELECT \* FROM ed\_deets;

-- Perform manipulation on simple view - insert, update, delete, drop

ALTER TABLE employee

MODIFY emp\_id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

MODIFY dept\_id INT DEFAULT 1,

MODIFY emp\_fname VARCHAR(20) DEFAULT NULL,

MODIFY emp\_lname VARCHAR(20) DEFAULT NULL,

MODIFY emp\_position VARCHAR(50) DEFAULT NULL,

MODIFY emp\_salary INT DEFAULT NULL,

MODIFY emp\_jdate DATE DEFAULT NULL

;

CREATE VIEW sv AS

SELECT

emp\_fname,

emp\_lname,

emp\_position

FROM

employee

ORDER BY

dept\_id

;

SELECT \* FROM sv;

INSERT INTO

sv

VALUES

("Shantanu", "Wable", "CTO"),

("Riya", "Wani", "CFO"),

("Atharva", "Zodpe", "GM")

;

DELETE FROM sv

WHERE emp\_fname = 'dolores';

UPDATE sv

SET emp\_lname = 'UPDATE'

WHERE emp\_fname = 'vero'

;

USE 31380\_db;

CREATE TABLE area (

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

radius INT,

area FLOAT

);

DELIMITER $$

CREATE PROCEDURE area\_proc (IN rad INT)

BEGIN

DECLARE a FLOAT DEFAULT 0.0;

SET a = 3.14 \* rad \* rad;

INSERT INTO area (radius, area) VALUES (rad, a);

END $$

DELIMITER ;

-- Calling the procedure

CALL area\_proc(5);

-- Viewing inserted data

SELECT \* FROM area;

/\*

+----+--------+------+

| id | radius | area |

+----+--------+------+

| 1 | 5 | 78.5 |

+----+--------+------+

\*/

-- Modified Procedure

DELIMITER $$

CREATE PROCEDURE mod\_area (IN rad INT)

BEGIN

DECLARE a FLOAT DEFAULT 0.0;

WHILE rad <= 9 DO

SET a = 3.14 \* rad \* rad;

INSERT INTO area (radius, area) VALUES (rad, a);

SET rad = rad + 1;

END WHILE;

END $$

DELIMITER ;

-- Calling modified procedure

CALL mod\_area (2);

-- Viewing inserted data

SELECT \* FROM area;

/\*

+----+--------+--------+

| id | radius | area |

+----+--------+--------+

| 1 | 5 | 78.5 |

| 2 | 2 | 12.56 |

| 3 | 3 | 28.26 |

| 4 | 4 | 50.24 |

| 5 | 5 | 78.5 |

| 6 | 6 | 113.04 |

| 7 | 7 | 153.86 |

| 8 | 8 | 200.96 |

| 9 | 9 | 254.34 |

+----+--------+--------+

\*/

USE 31380\_db;

CREATE TABLE stud\_marks(

name VARCHAR(50),

total\_marks INT

);

CREATE TABLE result (

roll INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

class VARCHAR(25) DEFAULT NULL,

name VARCHAR(50)

);

INSERT INTO stud\_marks VALUES

('aliquid',756),

('et',1253),

('ab',755),

('hic',1285),

('sunt',877),

('suscipit',696),

('eos',1404),

('necessitatibus',873),

('praesentium',1353),

('et',1154),

('officia',619),

('asperiores',871),

('illum',808),

('quis',1108),

('incidunt',1306),

('nemo',1595),

('et',705),

('soluta',783),

('ratione',821),

('ut',753)

;

DELIMITER $$

-- Function to classify marks

CREATE FUNCTION IF NOT EXISTS classify (marks INT) RETURNS VARCHAR(50)

DETERMINISTIC

BEGIN

IF marks BETWEEN 990 AND 1500 THEN

RETURN 'Distinction';

ELSEIF marks BETWEEN 900 AND 989 THEN

RETURN 'First Class';

ELSEIF marks BETWEEN 825 AND 899 THEN

RETURN 'Higher Second Class';

ELSE

RETURN 'Pass';

END IF;

END $$

-- Procedure to insert into result table

CREATE PROCEDURE proc\_grade (IN stud\_name VARCHAR(50), IN roll INT)

BEGIN

DECLARE clss VARCHAR(25);

DECLARE studmarks INT;

SELECT total\_marks INTO studmarks FROM stud\_marks WHERE name = stud\_name;

SET clss := classify(studmarks);

INSERT INTO result VALUES (roll, clss, stud\_name);

END $$

DELIMITER ;

CALL proc\_grade(900, 'Shantanu', 31380);

CALL proc\_grade(1000, 'Riya', 31381);

CALL proc\_grade(1490, 'Atharva', 31382);

CALL proc\_grade(799, 'Durvesh', 31379);

CALL proc\_grade(750, 'Rhea', 31364);

CALL proc\_grade(869, 'Ridhima', 31372);

CALL proc\_grade(500, 'Piyush', 31466);

CALL proc\_grade(1005, 'Riddhi', 31470);

SELECT \* FROM result;

/\*

+-------+-------------+----------+

| roll | class | name |

+-------+-------------+----------+

| 31364 | Pass | Rhea |

| 31379 | Pass | Durvesh |

| 31380 | First Class | Shantanu |

| 31381 | Distinction | Riya |

| 31382 | Distinction | Atharva |

| 31466 | Pass | Piyush |

| 31470 | Distinction | Riddhi |

+-------+-------------+----------+

7 rows in set (0.001 sec)

\*/

USE 31380\_db;

CREATE TABLE N\_EmpID (

emp\_id INT,

dept\_id INT,

emp\_fname VARCHAR(20),

emp\_lname VARCHAR(20),

emp\_position VARCHAR(50),

emp\_salary INT,

emp\_jdate DATE,

PRIMARY KEY (emp\_id)

);

CREATE TABLE O\_EmpID (

emp\_id INT,

dept\_id INT,

emp\_fname VARCHAR(20),

emp\_lname VARCHAR(20),

emp\_position VARCHAR(50),

emp\_salary INT,

emp\_jdate DATE,

PRIMARY KEY (emp\_id)

);

INSERT INTO O\_EmpID VALUES

(19,2,'molestiae','nostrum','quisquam',850067,'1990-02-16'),

(23,2,'aut','quis','harum',313554,'1976-09-16'),

(32,2,'et','et','architecto',942122,'1972-03-16'),

(33,3,'voluptatem','at','rerum',731520,'1990-06-28'),

(34,2,'fugit','harum','omnis',477917,'1985-08-06'),

(36,1,'provident','voluptatem','at',145306,'1995-07-06'),

(55,3,'suscipit','et','et',481782,'2011-05-10'),

(59,1,'ea','temporibus','qui',861072,'1988-07-26'),

(60,3,'fuga','quia','placeat',393128,'1988-10-12'),

(63,1,'voluptatem','nihil','reiciendis',856496,'1986-10-27'),

(67,3,'amet','expedita','similique',720147,'2000-11-24'),

(69,1,'dolore','magni','eius',414471,'1972-11-21'),

(70,2,'quos','veritatis','ex',938067,'1997-04-13'),

(81,2,'sequi','aut','voluptatibus',120184,'1987-12-17'),

(84,3,'ipsum','sunt','nemo',446727,'1977-03-24'),

(91,1,'tempore','qui','quia',338584,'1992-07-24'),

(110,1,'consequatur','ipsam','est',576805,'2001-04-22'),

(111,2,'eveniet','assumenda','rerum',526825,'2006-04-23'),

(123,3,'optio','esse','est',730407,'1989-06-18'),

(138,2,'nihil','rerum','eum',311204,'2009-03-02');

INSERT INTO N\_EmpID VALUES

(19,2,'molestiae','nostrum','quisquam',850067,'1990-02-16'),

(23,2,'aut','quis','harum',313554,'1976-09-16'),

(32,2,'et','et','architecto',942122,'1972-03-16'),

(33,3,'voluptatem','at','rerum',731520,'1990-06-28'),

(34,2,'fugit','harum','omnis',477917,'1985-08-06'),

(36,1,'provident','voluptatem','at',145306,'1995-07-06'),

(55,3,'suscipit','et','et',481782,'2011-05-10'),

(165,1,'molestiae','itaque','dolorem',948514,'1975-08-09'),

(168,2,'id','et','ut',771819,'2003-08-17'),

(172,1,'iure','omnis','ut',486159,'2006-09-16'),

(173,1,'perferendis','omnis','veniam',845958,'1990-07-14'),

(174,2,'alias','debitis','aut',679066,'2014-04-10'),

(179,2,'doloremque','voluptate','voluptate',148700,'1989-03-13'),

(184,1,'sed','error','reiciendis',993787,'1971-11-13'),

(186,1,'voluptas','eum','et',940556,'1983-09-08'),

(195,2,'nesciunt','non','autem',215182,'2011-04-22'),

(198,1,'pariatur','autem','deserunt',974698,'1972-04-24');

DELIMITER $$

CREATE PROCEDURE mergeEMP ()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE eno INT;

DECLARE old\_cursor CURSOR FOR SELECT emp\_id FROM O\_EmpID;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN old\_cursor;

getEmpID: LOOP

IF done = TRUE THEN

LEAVE getEmpID;

END IF;

FETCH old\_cursor INTO eno;

IF NOT EXISTS (SELECT 1 FROM N\_EmpID where emp\_id = eno) THEN

INSERT INTO N\_EmpID

SELECT \* FROM O\_EmpID

WHERE O\_EmpID.emp\_id = eno;

END IF;

END LOOP;

CLOSE old\_cursor;

END $$

DELIMITER ;

CALL mergeEMP();

DELIMITER $$

CREATE PROCEDURE mergeEMPwithinBounds (IN lb INT, IN ub INT)

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE eno INT;

DECLARE old\_cursor1 CURSOR FOR SELECT emp\_id FROM O\_EmpID WHERE emp\_id BETWEEN lb AND ub;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN old\_cursor1;

getEmpID: LOOP

IF done = TRUE THEN

LEAVE getEmpID;

END IF;

FETCH old\_cursor1 INTO eno;

IF NOT EXISTS (SELECT 1 FROM N\_EmpID where emp\_id = eno) THEN

INSERT INTO N\_EmpID

SELECT \* FROM O\_EmpID

WHERE O\_EmpID.emp\_id = eno;

END IF;

END LOOP;

CLOSE old\_cursor1;

END $$

DELIMITER ;

USE 31380\_db;

CREATE TABLE Library (

book\_id INT UNIQUE PRIMARY KEY NOT NULL AUTO\_INCREMENT,

book\_name VARCHAR(100) DEFAULT '',

isbn INT UNIQUE NOT NULL,

page\_count INT,

author VARCHAR(100),

year DATE,

copies\_sold INT

);

CREATE TABLE Library\_Audit (

update\_id INT UNIQUE PRIMARY KEY NOT NULL AUTO\_INCREMENT,

book\_id INT,

old\_book\_name VARCHAR(100) DEFAULT '',

old\_isbn INT UNIQUE NOT NULL,

old\_page\_count INT,

old\_author VARCHAR(100),

old\_year DATE,

old\_copies\_sold INT

);

INSERT INTO Library VALUES

(1,'totam',14448882,425,'expedita','2023-03-19',43849117),

(2,'voluptates',48822856,333,'sit','1974-08-27',11021007),

(3,'quae',10721240,165,'voluptates','2011-05-06',72946253),

(4,'impedit',77143556,133,'quam','1995-01-13',20492263),

(5,'cum',57158171,667,'ipsa','2000-01-26',29278743),

(6,'officia',23698255,804,'temporibus','2007-11-06',77006951),

(7,'rerum',52692996,871,'numquam','2012-08-23',23321626),

(8,'perspiciatis',80040097,697,'sunt','2006-05-02',80292082),

(9,'voluptate',78591954,642,'voluptatem','1990-07-11',23638319),

(10,'iure',37495366,644,'non','2020-03-08',59228850),

(11,'repellendus',29416436,841,'in','1982-03-28',63717696),

(12,'earum',14016995,778,'officia','1971-09-02',43033034),

(13,'quasi',46762666,855,'magni','1973-04-12',22185173),

(14,'nihil',12714950,583,'aspernatur','1991-07-24',83182555),

(15,'perspiciatis',86876793,590,'cum','1979-03-02',89074976),

(16,'qui',71512407,331,'maiores','2009-09-11',56650470),

(17,'neque',31651442,511,'perspiciatis','1980-11-07',18051664),

(18,'aut',77667078,386,'officia','1990-09-02',13554780),

(19,'qui',67772169,114,'expedita','2006-01-31',35066713),

(20,'quae',78427689,222,'quaerat','1992-07-22',54087575)

;

DELIMITER $$

CREATE TRIGGER update\_trig

BEFORE UPDATE ON Library

FOR EACH ROW

BEGIN

INSERT INTO Library\_Audit (book\_id, old\_book\_name, old\_isbn, old\_page\_count, old\_author, old\_year,old\_copies\_sold) VALUES

(OLD.book\_id, OLD.book\_name, OLD.isbn, OLD.page\_count, OLD.author, OLD.year, OLD.copies\_sold);

END $$

DELIMITER ;

-- Testing the trigger

UPDATE Library

SET book\_name = 'haha totam' WHERE book\_id = 1;

DELIMITER $$

CREATE TRIGGER error\_trig

BEFORE UPDATE ON Library

FOR EACH ROW

BEGIN

IF NEW.book\_name = OLD.book\_name THEN

SIGNAL SQLSTATE '45000'

SET message\_text = 'Same value of updated book';

END IF;

END $$

DELIMITER ;

UPDATE Library

SET book\_name = 'quae' WHERE book\_id = 3;

/\*