ASSIGNMENT 4

1. **List all the records in the student chart**

SYNTAX:

SELECT \* FROM STUDENTS;

1. **List the name surname and class of the student in the student table**

SYNTAX:

SELECT NAME,SURNAME,CLASS FROM STUDENTS;

1. **List the gender Female (F) records in the student table**

SYNTAX:

SELECT \* FROM STUDENTS WHERE GENDER='F';

1. **List the names of each class in the way of being seen once in the student table**

SYNTAX:

SELECT DISTINCT(CLASS) FROM STUDENTS;

SELECT CLASS FROM STUDENTS GROUP BY CLASS HAVING COUNT(CLASS)=1;

1. **List the students with Female gender and the class 10Math in the student table**

SYNTAX:

SELECT \* FROM STUDENTS WHERE GENDER='F' AND CLASS='10MATH';

1. **List the names, surnames and classes of the students in the class 10Math or 10Sci in the student table**

SYNTAX:

SELECT NAME,SURNAME,CLASS FROM STUDENTS WHERE CLASS='10MATH' OR CLASS='10SCI';

1. **List the students name surname and school number in the student table**

SYNTAX:

SELECT NAME,SURNAME,SCHOOL FROM STUDENTS;

1. **List the students name and surname by combining them as name surname in the student table**

SYNTAX:

SELECT CONCAT(NAME,' ',SURENAME) AS NAME\_OF\_STUDENT FROM STUDENTS;

1. **List the students with the names starting with “A” letter in the student table**

SYNTAX:

SELECT \* FROM STUDENTS WHERE NAME LIKE 'A%';

1. **List the book names and pages count with number of pages between 50 and 200 in the book table**

SYNTAX:

SELECT BNAME, BPAGECOUNT FROM BOOKS WHERE BPAGECOUNT BETWEEN 50 AND 200;

1. **List the students with names Emma, Sophia and Robert in the student table**

SYNTAX:

SELECT \* FROM STUDENTS WHERE NAME IN ('EMMA','SOPHIA','ROBERT');

1. **List the students with names starting with A D and K in the student table**

SYNTAX:

SELECT \* FROM STUDENTS WHERE NAME LIKE 'A%' OR NAME LIKE 'D%' OR NAME LIKE 'K%';

1. **List the names surnames classes and genders of males in 9Math or females in 9His in the student table**

SYNTAX:

SELECT NAME,SURNAME,CLASS, GENDER FROM STUDENTS WHERE (CLASS='9MATH' AND GENDER='M') OR (CLASS='9HIS' AND GENDER='F');

1. **List the males whose classes are 10Math or 10Bio**

SYNTAX:

SELECT \* FROM STUDENTS WHERE GENDER='M' AND (CLASS='10MATH' OR CLASS='10BIO');

1. **List the students with birth year 1989 in the student table**

SYNTAX:

SELECT \* FROM STUDENTS WHERE BIRTHDATE LIKE '1989%';

1. **List the female students with student numbers between 30 and 50**

SYNTAX:

SELECT \* FROM STUDENTS WHERE GENDER='F' AND STUDENTID BETWEEN 30 AND 50;

1. **List the students according to their names**

SYNTAX:

SELECT \* FROM STUDENTS ORDER BY NAME;

1. **List the students by names for those with same names. List them by their surnames**

SYNTAX:

SELECT NAME,SURENAME,COUNT(NAME) AS NUM\_DUP FROM STUDENTS GROUP BY NAME, SURENAME HAVING COUNT(NAME)>1;

1. **List the students in 10Math by decreasing school numbers**

SYNTAX:

SELECT \* FROM STUDENTS WHERE CLASS='10MATH' ORDER BY STUDENTID DESC;

1. **List the first 10 records in the student chart**

SYNTAX:

SELECT TOP 10 \* FROM STUDENTS;

1. **List the first 10 records name surname and date of birth information in the student table**

SYNTAX:

SELECT TOP 10 NAME,SURNAME,BIRTHDATE FROM STUDENTS;

1. **List the book with the most page number**

SYNTAX:

SELECT BNAME,BPAGECOUNT FROM BOOKS WHERE BPAGECOUNT=(SELECT MAX(BPAGECOUNT) FROM BOOKS);

1. **List the youngest student in the student table**

SYNTAX:

SELECT \* FROM STUDENTS WHERE BIRTHDATE=(SELECT MAX(BIRTHDATE) FROM STUDENTS);

1. **List the oldest student in the 10Math class**

SYNTAX:

SELECT \* FROM STUDENTS WHERE BIRTHDATE=(SELECT MIN(BIRTHDATE) FROM STUDENTS WHERE CLASS='10MATH');

1. **List the books with the second letter N**

SYNTAX:

SELECT \* FROM BOOKS WHERE BNAME LIKE '\_L%';

1. **List the students by grouping according to their classes**

Syntax:

SELECT NAME,CLASS FROM STUDENTS GROUP BY CLASS, NAME;

1. **List the students to be different in each questioning randomly**

UNABLE TO UNDERSTAND THE ASK

1. **Pick a random student from student table**

SYNTAX:

SELECT TOP 1 \* FROM STUDENTS ORDER BY NEWID();

1. **Bring some random student’s name, surname and number from class 10Math**

SYNTAX:

SELECT TOP 1 NAME,SURENAME,STUDENTID FROM STUDENTS WHERE CLASS='10MATH' ORDER BY NEWID();

1. **Add the writer named Smith Allen to the authors table**

SYNTAX:

INSERT INTO AUTHORS VALUES ('A\_006','SMITH','ALLEN');

1. **Add the genre of biography to the genre table**

SYNTAX:

INSERT INTO TYPESS VALUES ('T\_006','BIOGRAPHY');

1. **Add 10Math Class male named Thomas Nelson , 9Bio class female named Sally Allen and 11His Class female named Linda Sandra in one question**

SYNTAX:

INSERT INTO STUDENTS (STUDENTID,NAME, SURENAME,CLASS, GENDER) VALUES

('S\_009','THOMAS','NELSON','10MATH','M'),

('S\_010','SALLY','ALLEN','9BIO','F'),

('S\_011','LINDA','SANDRA','11HIS','F');

1. **Add a random student in the students chart to the writers chart as an authors**

SYNTAX:

INSERT INTO AUTHORS (A\_NAME,SURNAME)

SELECT TOP 1 NAME,SURNAME FROM STUDENTS ORDER BY NEWID();

1. **Add students with student numbers between 10 and 30 as authors**

SYNTAX:

INSERT INTO AUTHORS (A\_NAME,SURENAME)

(SELECT NAME,SURNAME FROM STUDENTS WHERE STUDENTID BETWEEN 10 AND 30);

1. **Add the writer named Cindy Brown and make him write his writer number (Note: The last increased rate in automatic enhancing is hold in @@IDENTITY factor)**

SYNTAX:

SET IDENTITY\_INSERT AUTHORS ON;

INSERT INTO AUTHORS VALUES (‘A\_009’, ‘CINDY’, ‘BROWN’);

SET IDENTITY\_INSERT AUTHORS OFF;

1. **Change the class of the student whose school number is 3 from 10Bio to 10His**

SYNTAX:

UPDATE STUDENTS SET CLASS='10HIS' WHERE STUDENTID='S\_003';

1. **Transfer all the students in 9Math Class to 10Math Class**

SYNTAX:

UPDATE STUDENTS SET CLASS='10MATH’ WHERE CLASS= ‘9MATH’;

1. **Increase all of the students’ score by 5 points**

SYNTAX:

UPDATE STUDENTS SET POINTS=POINTS+5;

1. **Delete the author #25**

SYNTAX:

DELETE FROM AUTHORS WHERE AUTHORID LIKE ‘%25’;

1. **List the students whose birth dates are null**

SYNTAX:

SELECT \* FROM STUDENTS WHERE BIRTHDATE IS NULL;

1. **List the name ,surname and the dates of received books of the student**

SYNTAX:

SELECT S.NAME,S.SURENAME,BO.BROUGHTDATE FROM STUDENTS S JOIN BORROWS BO ON S.STUDENTID=BO.STUDENTID ORDER BY NAME;