Lei Garay

SQL Assignment

Section 1 & 2

CREATE TABLE DEPARTMENT

(

DEPARTMENT\_ID NUMBER NOT NULL,

DEPARTMENT\_NAME VARCHAR(30) ,

DEPARTMENT\_BLOCK\_NUMBER NUMBER ,

PRIMARY KEY(DEPARTMENT\_ID)

);

CREATE TABLE STUDENT

(

STUDENT\_ID NUMBER NOT NULL,

STUDENT\_NAME VARCHAR(30),

STUDENT\_BLOCK\_NUMBER VARCHAR(40),

ADDRESS VARCHAR(30),

CITY VARCHAR(30),

DEPARTMENT\_ID NUMBER,

PRIMARY KEY (STUDENT\_ID),

FOREIGN KEY (DEPARTMENT\_ID) REFERENCES DEPARTMENT(DEPARTMENT\_ID)

);

CREATE TABLE STAFF

(

STAFF\_ID NUMBER NOT NULL,

STAFF\_NAME VARCHAR(30),

DEPARTMENT\_ID NUMBER,

PRIMARY KEY(STAFF\_ID),

FOREIGN KEY(DEPARTMENT\_ID) REFERENCES DEPARTMENT(DEPARTMENT\_ID)

);

CREATE TABLE SUBJECT

(

SUBJECT\_ID NUMBER NOT NULL,

SUBJECT\_NAME VARCHAR(30),

SUBJECT\_CODE VARCHAR(10),

STAFF\_ID NUMBER,

PRIMARY KEY (SUBJECT\_ID),

FOREIGN KEY (STAFF\_ID) REFERENCES STAFF(STAFF\_ID)

);

CREATE TABLE MARK

(

VALUE NUMBER NOT NULL,

SUBJECT\_ID NUMBER,

STUDENT\_ID NUMBER,

PRIMARY KEY (SUBJECT\_ID, STUDENT\_ID),

FOREIGN KEY (SUBJECT\_ID) REFERENCES SUBJECT(SUBJECT\_ID),

FOREIGN KEY (STUDENT\_ID) REFERENCES STUDENT(STUDENT\_ID)

);

ALTER TABLE STAFF MODIFY (STAFF\_NAME NOT NULL);

ALTER TABLE STUDENT ADD (EMAILID VARCHAR(20));

ALTER TABLE STUDENT MODIFY (EMAILID VARCHAR(50));

ALTER TABLE STUDENT DROP COLUMN EMAILID;

Error report -

SQL Error: ORA-12988: cannot drop column from table owned by SYS

12988. 00000 - "cannot drop column from table owned by SYS"

\*Cause: An attempt was made to drop a column from a system table.

\*Action: This action is not allowed

Section 3-

Updating Records

12 - Update a record by writing a query to update the subject\_name in the subject table from Sales to Computer Science and subject\_code from 1842 to 1919.

UPDATE SUBJECT

SET SUBJECT\_NAME = 'Computer Science', SUBJECT\_CODE = 1919

WHERE SUBJECT\_NAME = 'Sales' and SUBJECT\_CODE = 1842;

Section 4-

Deleting Records

13 - Delete the row from the subject table where subject name is Accounting by writing the appropriate query.

DELETE

FROM SUBJECT

WHERE SUBJECT\_NAME = 'Accounting';

Error starting at line : 1 in command -

Delete from subject

where subject\_name = 'Accounting'

Error report -

SQL Error: ORA-02292: integrity constraint (SYS.SYS\_C008446) violated - child record found

02292. 00000 - "integrity constraint (%s.%s) violated - child record found"

\*Cause: attempted to delete a parent key value that had a foreign

dependency.

\*Action: delete dependencies first then parent or disable constraint.

Section 5-

Basic Selection of Records

14 - Display the names of the department in the college by writing the appropriate query. Please note that these must be displayed in ascending order.

SELECT DEPARTMENT\_NAME

FROM DEPARTMENT

ORDER BY DEPARTMENT\_NAME ASC;

15- Display the names of the departments where departments block number is between 3 and 10 by writing the appropriate query.

SELECT DEPARTMENT\_NAME

FROM DEPARTMENT

WHERE DEPARTMENT\_BLOCK\_NUMBER

BETWEEN 3 AND 10;

16- Display the names of all the students in the college by writing the appropriate query. Please note these must be displayed in ascending order.

SELECT STUDENT\_NAME

FROM STUDENT

ORDER BY STUDENT\_NAME ASC;

Section 6-

Selecting Single Rows

17- Display the names of the students who are from Chicago, Taylor and San Jose.

Please note these must be displayed in ascending order of their respective id.

SELECT STUDENT\_ID, STUDENT\_NAME

FROM STUDENT

WHERE CITY

IN ('Chicago', 'Taylor','San Jose')

ORDER BY STUDENT\_ID;

18- Writing the correct query, display the address and city of the students table give the alias as Address\_Student.

SELECT CITY, ADDRESS

|| ADDRESS AS STUDENT\_ADDRESS

FROM STUDENT;

19- Display all of the student’s names whose names are of 6 characters in length by writing the correct query.

SELECT STUDENT\_NAME

FROM STUDENT

WHERE LENGTH (STUDENT\_NAME) = 6;

Section 7-

Selecting Groups

20- Display the blocknumber and number of departments in each block by writing the correct query that is ordered by block id. Make sure it is displayed as count (department\_name)

SELECT COUNT(DEPARTMENT\_BLOCK\_NUMBER)

FROM DEPARTMENT; FIX THIS

21- Display the number of students in the college by writing the correct query and give an alias as stud\_count.

SELECT COUNT(STUDENT\_NAME)

AS STUD\_COUNT

FROM STUDENT;

Section 8-

SQL Joins

22- Display the names of the department and the student count in each department by writing the correct query. The student count in each department must be in ascending order based on the department name and an alias of student\_count for the student count.

SELECT DEPARTMENT\_NAME

AS DEPARTMENT\_NAME,

COUNT(\*)

AS STUDENT\_COUNT

FROM DEPARTMENT

INNER

JOIN STUDENT

ON STUDENT.DEPARTMENT\_ID = DEPARTMENT.DEPARTMENT\_ID

GROUP BY DEPARTMENT.DEPARTMENT\_ID, DEPARTMENT\_NAME

ORDER BY DEPARTMENT.DEPARTMENT\_NAME;

23- Display the Student\_Name from STUDENT and the Subject\_name from SUBJECT where the Subject\_code from SUBJECT is greater than 1600.

SELECT STUDENT\_NAME, SUBJECT.SUBJECT\_NAME

FROM

(

SELECT STUDENT\_NAME

FROM STUDENT

GROUP BY STUDENT\_NAME

)

JOIN SUBJECT

ON SUBJECT.SUBJECT\_CODE > 1600;

24- Display the Stundent\_Name from STUDENTS and the Subject\_name from SUBJECT where the value on MARK table is less 3.

SELECT STUDENT\_NAME,SUBJECT.SUBJECT\_NAME

FROM STUDENT

INNER

JOIN MARK

ON MARK.STUDENT\_ID = STUDENT.STUDENT\_ID

INNER

JOIN SUBJECT

ON MARK.SUBJECT\_ID = SUBJECT.SUBJECT\_ID

WHERE MARK.VALUE = 3;

Section 9 –

Selecting Sub-Queries

25- Display the block number in which the maximum number of departments is located by writing the correct sub-query.

SELECT DEPARTMENT\_NAME,

DEPARTMENT\_BLOCK\_NUMBER

FROM DEPARTMENT

WHERE DEPARTMENT\_NAME = (

SELECT MAX(DEPARTMENT\_NAME)

FROM DEPARTMENT

);

26- Display the names of the staff who are not handling any subjects by ascending order using the correct sub-query.

SELECT STAFF\_NAME,

STAFF\_ID

FROM STAFF

WHERE STAFF\_ID NOT IN

(

SELECT STAFF\_ID

FROM SUBJECT

);

Secction 10-

Functions

27- Write a function that takes department\_id as the input and returns the department\_name.

Use the function name below:

Function name: find\_dept\_name

CREATE OR REPLACE FUNCTION

FIND\_DEPT\_NAME(X NUMBER)

RETURN VARCHAR2

IS DEP VARCHAR2(30);

BEGIN

SELECT DEPARTMENT\_NAME

INTO DEP

FROM DEPARTMENT

WHERE DEPARTMENT\_ID =X;

RETURN (DEP);

END;

28- Write a function that takes department id as the input and returns the block number.

Use the function name below:

Function name: find\_dept\_block

CREATE OR REPLACE FUNCTION

FIND\_DEPT\_BLOCK(X NUMBER)

RETURN NUMBER

IS DEPNUM NUMBER;

BEGIN

SELECT DEPARTMENT\_BLOCK\_NUMBER

INTO DEPNUM

FROM DEPARTMENT

WHERE DEPARTMENT\_ID =X;

RETURN (DEPNUM);

END;

29- Write a function that takes the staff id as the input and returns the staff name.

Use the function name below:

Function name: find\_staff\_name

CREATE OR REPLACE FUNCTION

FIND\_DEPT\_NAME(X NUMBER)

RETURN VARCHAR2

IS DEP VARCHAR2(30);

BEGIN

SELECT DEPARTMENT\_NAME

INTO DEP

FROM DEPARTMENT

WHERE DEPARTMENT\_ID =X;

RETURN (DEP);

END;

Section 11

Triggers

30 Create a trigger with the name 'trigger\_department\_af\_update' which will display “DEPARTMENTS table has been updated”

after an attempt to update the DEPARTMENTS has been made.

CREATE OR REPLACE TRIGGER

TRIGGER\_DEPARTMENT\_AF\_UPDATE

AFTER DELETE

ON DEPARTMENT

FOR EACH ROW

BEGIN

TRIGGER\_DEPARTMENT\_AF\_UPDATE;

DBMS\_OUTPUT.put\_line('DEPARTMENTS table has been updated' );

END;

Error report -

ORA-04089: cannot create triggers on objects owned by SYS

04089. 00000 - "cannot create triggers on objects owned by SYS"

\*Cause: An attempt was made to create a trigger on an object owned by SYS.

\*Action: Do not create triggers on objects owned by SYS.

Trigger name: trigger\_department\_af\_update

31 Create a trigger with the name ‘trigger\_department\_bf\_delete’ which will display “A row has been deleted from DEPARTMENT”

before an attempt to delete a row is execute on DEPARTMENT.

Trigger name : trigger\_department\_bf\_delete

CREATE OR REPLACE TRIGGER

TRIGGER\_DEPARTMENT\_BF\_UPDATE

AFTER DELETE

ON DEPARTMENT

FOR EACH ROW

BEGIN

TRIGGER\_DEPARTMENT\_BF\_UPDATE;

DBMS\_OUTPUT.put\_line('A row has been deleted from DEPARTMENT' );

END;

Section 12

Views and Index

32 Create an Index command that will reference all of the students names containing the letter ‘b’ on the Student table.

CREATE INEX IDX\_STUDENTS

ON STUDENT(STUDENT\_NAME);

33 Create a view from the staff table that will display staff names.

CREATE VIEW VIEW\_STAFF AS

SELECT STAFF\_NAME

FROM STAFF;

Cursors

36 Declare an explicit cursor using the STUDENT table to select of column. Fetch the rows using a loop and display each data retrieved.

DECLARE

nameholder VARCHAR2(30);

CURSOR CL1 IS

SELECT STUDENT\_NAME

FROM STUDENT;

BEGIN

OPEN CL1;

LOOP

FETCH CL1 INTO nameholder;

dbms\_output.put\_line (nameholder);

EXIT WHEN CL1%NOTFOUND;

END LOOP;

CLOSE CL1;

END;

Section 14

PACKAGES AND PROCEDURES

37 Define a package give it the name of College.

The definition should include a procedure give it the name of select\_departments.

It should also include a function that takes a argument of type number and returns a variable of type VARCHAR, give it the name of select\_student.

38 Implement the body of College department. the select\_departments procedure should display all column from DEPARTMENT.

The select\_student function should take the id of the student and return that student name.

create or replace PACKAGE COLLEGE AS

PROCEDURE SELECT\_DEPARTMENTS(DEPTID NUMBER);

FUNCTION SELECT\_STUDENT(X NUMBER)

RETURN VARCHAR2

IS NAMEHOLDER VARCHAR(30);

BEGIN

SELECT STUDENT\_NAME

INTO NAMEHOLDER

FROM STUDENT

WHERE STUDENT\_ID = ID;

RETURN (NAMEHOLDER);

END;

END COLLEGE;