Orcl5\_17

Name: DatabaseOracleJan20

1. Common questions about what is DDL. PL/SQL (same concepts in Microsoft SQL and how they are applied on Oracle)
2. For Lab1A, create table process as found in printed documents
3. Slide 13 (important)
4. Oracle naming constraints
5. There are things you can’t use in names
6. All oracle names begin with alpha character
7. It cannot be an sql reserve word
8. Check slide 15, about char and varchar
9. Bottom of slide 14, gender char.
10. Turning to numbers on Oracle (slide 16) to do with precision and scale)
11. So go through numbers & string (char & varchar) and how they are treated in Oracle
12. How we treat integers too
13. Slide 20 about date in oracle
14. Oracle date format
15. Constraints on oracle (integrity and value)
16. Integrity are just PK & FK
17. Value?
18. How null values are treated
19. Slide 25, you don’t delete a table, you drop it. Goes into a recycle bin
20. ‘purge’
21. Sequence slide 29 (not the same as @@identity in MS SQL
22. Slide 30, syntax, if you start with a +ve number, you increment by a +ve number
23. No cache
24. Slide 31
25. Slide 32 scripts
26. When creating a table, make sure you name your constraint (check on the slide on naming convection for constraint name)
27. Look at your ‘in’ operator in the slides
28. CREATE TABLE MM\_STUDENT
29. ( SID NUMBER (8.,0) CONSTRAINT PK\_MMSTUDENT\_SID PRIMARY KEY
30. CONSTRAINT NN\_MMSTUDENT\_SID NOT NULL,
31. SNAME VARCHAR2(50) DEFAULT ‘UNKNOWN’
32. CONSTRAINTS N\_MMSTUDENT\_SNAME NULL,
33. GENDER CHAR(1) CONSTRAINT CK\_MMSTUDENT\_GENDER\_MFN
34. CHECK(GENDER IN (‘M’, ‘F’, ‘N’))
35. CONSTRAINT NN\_MMSTUDENT\_GENDER NOT NULL,
36. EDATE DATE DEFAULT SYSDATE
37. CONSTRAINT NN\_MMSTUDENT\_EDATE NOT NULL
38. );

20200110 Class

Regular expressions (Check the file regular expressions on moodle)

For example, Microsoft regular expression might be Postal\_Code like ‘[a-z][0-9]….’ While Oracle is REGEXP\_LIKE(POSTAL\_CODE,’[A-Z][0-9]…’)

Hint; characters for start ^ and the $ for end are important

Read this about creating a table with both foreign key and primary key

create table MM\_student

( sid number(8,0) Constraint PK\_mmstudent\_sid PRIMARY KEY

Constraint NN\_mmstudent\_sid NOT NULL,

sname varchar2(50) default 'UNKNOWN'

Constraint N\_mmstudent\_sname NULL,

gender char(1) Constraint CK\_mmstudent\_gender\_MFN

Check(gender in ('M', 'F', 'N'))

Constraint NN\_mmstudent\_gender NOT NULL,

edate date default sysdate

Constraint NN\_mmstudent\_edate NOT NULL

);

create table MM\_course

( cid char(8) Constraint PK\_mmcourse\_cid PRIMARY KEY

Constraint NN\_mmcourse\_cid NOT NULL,

cname varchar2(50)

Constraint N\_mmcourse\_cname NULL,

location varchar2(20)

Constraint N\_mmcourse\_location NULL,

ccost number(6,2) default '575.00'

Constraint N\_mmcourse\_ccost NULL

);

create table MM\_grade

( sid number(8,0) Constraint FK\_MMGRADE\_SID FOREIGN KEY REFERENCES MMSTUDENT (SID)

CONSTRAINT NN\_MMGRADE\_SID NOT NULL,

CID CHAR(8) CONSTRAINT FK\_MMCOURSE\_CID FOREIGN KEY REFERENCES MMCOURSE (CID)

CONSTRAINT NN\_MMGRADE\_SID NOT NULL,

CONSTRAINT MMGrade\_SID\_CID\_PK PRIMARY KEY (SID, CID)

MARK NUMBER(5,2) Constraint CK\_mmGRADE\_MARK\_

CHECK ((MARK > 0) AND (MARK <100))

CONSTRAINT NN\_MMGRADE\_MARK NOT NULL

);

Class 20200113

1. Create stored procedure NB: This procedure has no error checks so no error message for this example. P stands for parameter

CREATE OR REPLACE PROCEDURE PR\_UPDATE\_MARK1

(P\_SID NUMBER, P\_CID CHAR, P\_MARK NUMBER)

AS

BEGIN

UPDATE MM\_GRADE

SET MARK = P\_MARK

WHERE SID = P\_SID AND

CID = P\_CID;

END PR\_UPDATE\_MARK1;

/

SHOW ERRORS;

1. Slide 7 (B-2) there is a command called SHOW ERROR; which shows some errors
2. ANOTHER procedure
3. NOTE BELOW HOW INPUT PARAMETERS ARE TREATED VS HOW LOCAL VARIABLES ARE TREATED
4. CREATE OR REPLACE PROCEDURE PR\_UPDATE\_MARK2

(P\_SNAME VARCHAR2, P\_CNAME VARCHAR2, P\_MARK NUMBER)

AS

V\_SID NUMBER(8,0);

V\_CID CHAR(8);

BEGIN

SELECT SID

INTO V\_SID

FROM MM\_STUDENT

WHERE SNAME = P\_SNAME;

SELECT CID

INTO V\_CID

FROM MM\_COURSE

WHERE CNAME = P\_CNAME;

UPDATE MM\_GRADE

SET MARK = P\_MARK

WHERE SID = V\_SID AND

CID = V\_CID;

END PR\_UPDATE\_MARK2;

/

SHOW ERRORS;

1. How do you run this procedure? Slide 18 has ‘EXEC PR\_ADD\_... (example of an execute command)
2. Can the procedure above be done without the local variables? Yes, by using subqueries
3. A subquery is a select! Does not have the where clause
4. But function might cause errors. Not advised

20200117

1. SQL PLUS; Used for formatting your data from SQL to a text editor (like how you will do in lab 1b) each line below on the code is line by line to show how you can format your data
2. Try this
3. Set Pagesize 99

Set Linesize 80

COLUMN Portfolio\_Number Format A9 Heading “Portfolio/Number”

COLUMN transaction)date format a20 heading “Transaction/Date”

COLUMN price\_per\_share format a10 heading “Price/Per/Share”

COLUMN exchange\_code format a8 heading “exchange/code”

COLUMN quantity format a8 heading “Quantity”

Select Portfolio\_Number,

To\_char(transaction\_date, ‘yyyy-mon-dd hh24:mi:ss’) Transaction\_Date,

Stock\_code, exchange code,

Broker\_number, Buy\_sell, quantity, to\_char(Quantity, ‘9,999,999’) Quantity,

To\_char(price\_per\_share, “$9,999.99’) Price\_per\_share

From transaction

Where Portfolio\_Number>535

1. Play with 80 & 99 to manage how your data will be displayed
2. HINT: The where clause in Lab 1b is crucial