***Please check for errors and possible additional notes ☺ required for the exma.***

**Final Exam COMP3610 Syntax for Oracle SQL and PL/SQL**

CREATE **PROCEDURE** *proc\_name*

(*p-parameter datatype*) IS

**CURSOR** *cursor\_name* IS

select *statment*;

*v\_variable datatype := initialvalue;*

BEGIN

FOR *record\_name* IN *cursor\_name*

loop

……

DBMS\_OUTPUT.PUT\_LINE (‘abc’ || *record\_name.column\_name*);

end loop;

END;

CREATE **TRIGGER** *trigger\_name*

BEFORE | AFTER |INSTEAD OF

INSERT | DELETE | UPDATE ON *tablename*

[FOR EACH [ROW| STATEMENT]]

WHEN *condition*

[DECLARE]

BEGIN

…..  
END;

CREATE **FUNCTION** *function\_name*

(*parameter\_name datatype, …*)

RETURN *datatype*  IS

*v\_variable datatype := initialvalue;*

BEGIN

……

RETURN expressionorvalue;

END;

CREATE **TYPE** *type\_name* AS **OBJECT**

(*attribute\_name* datatype, ….);

**SQL FUNCTIONS FOR STRING/CHARACTER**

**SUBSTR**(‘Exam’, 1, 2) returns ‘Ex’

**UPPER, LOWER, INITCAP, LENGTH**

**LPAD** (‘AAA’, 5, ‘\*’) returns \*\*AAA

**RPAD (**‘AAA’, 5, ‘\*’) returns AAA\*\*

**LENGTH(RTRIM('data '))** returns 4

**CONCAT(s1,s2) , s1 || s2** returns string s1s2

**TO\_CHAR**(SYSDATE, ‘YYYY-MM-DD’)

returns today’s date: ‘2016-12-16’

**REGEXP\_COUNT** ('AGGAGGTTTTT', 'A',1,'i') returns COUNT of ‘A’ starting from the 1st position of the string 'AGGAGGTTTTT' case insensitive (‘i’)

**SQL AGGREGATE (GROUP) FUNCTIONS**

**AVG(salary)** returns average value (ignores NULL values)

**SUM(salary)** returns sumof all salaries

**MIN(**column**), MAX(**column**),**

**COUNT (\* | [DISTINCT]** *columnname***)** returns # of rows containing a value (or distinct values) in the column

**SQL DATE/TIME FUNCTIONS**

**TO\_DATE(‘2014-04-15’, ‘YYYY-MM-DD’)**

**MONTHS\_BETWEEN** (d1,d2)

**ADD\_MONTHS** (d1, integer)

**LAST\_DAY (sysdate)** returns last day of current month

**TRUNC(sysdate,’MM’)** returns current month

**TO\_CHAR(sysdate,’D’)** returns one-digit day of the week

**EXTRACT(YEAR from SYSDATE)** returns numeric 2016

**TO\_DATE('2016-12-16 12:12', 'YYYY-MM-DD HH24:MI')** returns date

**SQL NUMBER FUNCTIONS**

**TO\_NUMBER, FLOOR, CEIL, LEAST, GREATEST,**

**ROUND (999.99, 0)** returns 1000

**NVL(v, 0)** returns v or 0 when v IS NULL

**NVL2 (v, a, b)** returns a when v IS NOT NULL, returns b when v IS NULL

**DML STATEMENTS**

INSERT INTO *tablename* [ (colname1, …)] VALUES (value1,…);

INSERT INTO tablename [ (colname1, …)]

SELECT ….;

UPDATE *tablename*

SET colname1 = value1 [,colname2 =value2, …]

[WHERE condition]

DELETE FROM *tablename*

[WHERE condition]

**JOIN TWO TABLES also LEFT [RIGHT] OUTER JOIN**

Patient (Patient\_id, DOB)

Medication(Patient\_id, Drug\_number, Start\_date, End\_date)

SELECT P.Patient\_id, M.Drug\_number

FROM Patient P, Medication M

WHERE P.Patient\_id = M.Patient\_id;

**- -Alternative syntax**

SELECT P.Patient\_id, M.Drug\_number

FROM Patient P **JOIN** Medication M

**ON** P.Patient\_id = M.Patient\_id;

**DDL STATEMENTS**

CREATE **TABLE** *tablename*

(*columnname* datatype [PRIMARY KEY], …

-- virtual column

c*olumnname* **AS** expression,

-- table constraint

CONSTRAINT *consname* CHECK(*colname* IN (*val1,val2*)) );

CREATE **VIEW** *viewname* **(***columname,…***)**

AS SELECT statement;

CREATE **INDEX** indexname ON tablename**(***columname,…***)** CREATE **SEQUENCE** *seqname* [INCREMENT BY value]

[START WITH value];

ALTER TABLE *tablename* ADD(*columname1 datatype*, …);