**Oracle POC**

* **CREATE TABLE Statement**

create table toys (

toy\_name varchar2(100),

colour varchar2(10),

price number(10, 2)

);

* **INSERT Statement**

insert into toys values ( 'Sir Stripypants', 'red', 0.01 );

insert into toys values ( 'Miss Smelly\_bottom', 'blue', 6.00 );

insert into toys values ( 'Cuteasaurus', 'blue', 17.22 );

insert into toys values ( 'Mr Bunnykins', 'red', 14.22 );

insert into toys values ( 'Baby Turtle', 'green', null );

* **SELECT Statement**

select \* from toys;

select toy\_name, price from toys;

* **UPDATE Statement**

UPDATE toys

SET toy\_name = 'Panda'

WHERE colour = ‘green’;

* **DELETE Statement**

DELETE FROM toys

WHERE colour = ‘red’;

* **Filtering Data**

select \* from toys

where toy\_name = 'Sir Stripypants';

select \* from toys

where colour = 'red';

* **LIKE Clause**

SELECT \* FROM toys

WHERE toy\_name LIKE 'Ba%';

* **ORDER BY Clause**

SELECT \* FROM toys

ORDER BY toy\_name, price

* **Joining Tables**

**Inner Join**

SELECT suppliers.supplier\_id, suppliers.supplier\_name, orders.order\_date

FROM suppliers

INNER JOIN orders

ON suppliers.supplier\_id = orders.supplier\_id;

**Left Join**

SELECT suppliers.supplier\_id, suppliers.supplier\_name, orders.order\_date

FROM suppliers

LEFT OUTER JOIN orders

ON suppliers.supplier\_id = orders.supplier\_id;

**Right Join**

SELECT orders.order\_id, orders.order\_date, suppliers.supplier\_name

FROM suppliers

RIGHT OUTER JOIN orders

ON suppliers.supplier\_id = orders.supplier\_id;

* **Sub-Queries**

SELECT suppliers.name, subquery1.total\_amt

FROM suppliers,

(SELECT supplier\_id, SUM(orders.amount) AS total\_amt

FROM orders

GROUP BY supplier\_id) subquery1

WHERE subquery1.supplier\_id = suppliers.supplier\_id;

* **Stored Procedure**

CREATE PROCEDURE remove\_emp (employee\_id NUMBER) AS

tot\_emps NUMBER;

BEGIN

DELETE FROM employees

WHERE employees.employee\_id = remove\_emp.employee\_id;

tot\_emps := tot\_emps - 1;

END;

* **Transactions**

START TRANSACTION;

SELECT @A:=SUM(salary) FROM table1 WHERE type=1;

UPDATE table2 SET summary=@A WHERE type=1;

COMMIT;

* **Indexes**

CREATE INDEX upper\_ix ON employees (UPPER(last\_name));

* **Functions**

create or replace function adder(n1 in number, n2 in number)

return number

is

n3 number(8);

begin

n3 :=n1+n2;

return n3;

end;

**PL/SQL Block**

**For Loop:**

DECLARE

x NUMBER := 100;

BEGIN

FOR i IN 1..10 LOOP

IF MOD(i,2) = 0 THEN -- i is even

INSERT INTO temp VALUES (i, x, 'i is even');

ELSE

INSERT INTO temp VALUES (i, x, 'i is odd');

END IF;

x := x + 100;

END LOOP;

COMMIT;

END;

**Cursor:**

DECLARE

CURSOR c1 is

SELECT ename, empno, sal FROM emp

ORDER BY sal DESC; -- start with highest paid employee

my\_ename VARCHAR2(10);

my\_empno NUMBER(4);

my\_sal NUMBER(7,2);

BEGIN

OPEN c1;

FOR i IN 1..5 LOOP

FETCH c1 INTO my\_ename, my\_empno, my\_sal;

EXIT WHEN c1%NOTFOUND; /\* in case the number requested \*/

/\* is more than the total \*/

/\* number of employees \*/

INSERT INTO temp VALUES (my\_sal, my\_empno, my\_ename);

COMMIT;

END LOOP;

CLOSE c1;

END;

**Scoping:**

DECLARE

x NUMBER := 0;

counter NUMBER := 0;

BEGIN

FOR i IN 1..4 LOOP

x := x + 1000;

counter := counter + 1;

INSERT INTO temp VALUES (x, counter, 'in OUTER loop');

/\* start an inner block \*/

DECLARE

x NUMBER := 0; -- this is a local version of x

BEGIN

FOR i IN 1..4 LOOP

x := x + 1; -- this increments the local x

counter := counter + 1;

INSERT INTO temp VALUES (x, counter, 'inner loop');

END LOOP;

END;

END LOOP;

COMMIT;

END;