**Yellow – Create table code**

**Red – Trigger code**

**CARS table:**

CREATE TABLE "CARS"

( "CAR\_ID" NUMBER,

"CAR\_MODEL" VARCHAR2(50),

"CAR\_COLOR" VARCHAR2(50),

"REL\_DATE" VARCHAR2(50),

CONSTRAINT "CARS\_CONPO" PRIMARY KEY ("CAR\_ID")

USING INDEX ENABLE

) ;

CREATE OR REPLACE EDITIONABLE TRIGGER "AUTOID"

before insert ON Cars

for each row

declare

max\_n number;

begin

if :NEW.car\_id is null then

select max(car\_id) into max\_n from Cars;

max\_n := max\_n + 1;

:NEW.car\_id := max\_n;

end if;

end;

/

ALTER TRIGGER "AUTOID" ENABLE;

**CUSTOMERS\_C table:**

CREATE TABLE "CUSTOMERS\_C"

( "CUST\_ID" NUMBER,

"CAST\_NAME" VARCHAR2(50),

"CAST\_PHONE" VARCHAR2(50),

"CAST\_ADDRESS" VARCHAR2(50),

CONSTRAINT "CUSTOMERS\_C\_CONPO" PRIMARY KEY ("CUST\_ID")

USING INDEX ENABLE

) ;

**CREATE OR REPLACE EDITIONABLE TRIGGER "AUTOID\_CUST"**

**before insert ON Customers\_c**

**for each row**

**declare**

**max\_n number;**

**begin**

**if :NEW.cust\_id is null then**

**select max(cust\_id) into max\_n from Customers\_c;**

**max\_n := max\_n + 1;**

**:NEW.cust\_id := max\_n;**

**end if;**

**end;**

**/**

**ALTER TRIGGER "AUTOID\_CUST" ENABLE;**

**EMPLOYEES\_C table:**

CREATE TABLE "EMPLOYEES\_C"

( "EMP\_ID" NUMBER,

"EMP\_NAME" VARCHAR2(50),

"EMP\_PHONE" VARCHAR2(50),

"EMP\_SALARY" NUMBER,

CONSTRAINT "EMPLOYEES\_C\_CONPO" PRIMARY KEY ("EMP\_ID")

USING INDEX ENABLE

) ;

CREATE OR REPLACE EDITIONABLE TRIGGER "AUTOID\_EMPLOYEE"

before insert ON Employees\_c

for each row

declare

max\_n number;

begin

if :NEW.emp\_id is null then

select max(emp\_id) into max\_n from Employees\_c;

max\_n := max\_n + 1;

:NEW.emp\_id := max\_n;

end if;

end;

/

ALTER TRIGGER "AUTOID\_EMPLOYEE" ENABLE;

CREATE OR REPLACE EDITIONABLE TRIGGER "SALARY\_REPORT"

after insert or update or delete ON Employees\_c

for each row

declare

pragma autonomous\_transaction;

sal number;

inc number;

ove number;

begin

APEX\_UTIL.PAUSE(5);

Select SUM(orders\_c.order\_cost) into inc from orders\_c, orders\_consult where orders\_consult.status = 1 and orders\_c.order\_id = orders\_consult.order\_id;

inc := (inc \* 4) / 10;

Select SUM(emp\_salary) into sal from Employees\_c;

ove := inc - sal - 40000;

Insert into Income values(Sysdate, sal, inc, 40000, ove);

COMMIT;

end;

/

ALTER TRIGGER "SALARY\_REPORT" ENABLE;

**ORDERS\_C table:**

CREATE TABLE "ORDERS\_C"

( "ORDER\_ID" NUMBER,

"CUST\_ID" NUMBER,

"CAR\_ID" NUMBER,

"ORDER\_COST" NUMBER,

"ORDER\_DATE" DATE,

CONSTRAINT "ORDERS\_C\_CONP" PRIMARY KEY ("ORDER\_ID")

USING INDEX ENABLE

) ;

ALTER TABLE "ORDERS\_C" ADD CONSTRAINT "ORDERS\_C\_CONFC" FOREIGN KEY ("CUST\_ID")

REFERENCES "CUSTOMERS\_C" ("CUST\_ID") ON DELETE CASCADE ENABLE;

ALTER TABLE "ORDERS\_C" ADD CONSTRAINT "ORDERS\_C\_CONFCAR" FOREIGN KEY ("CAR\_ID")

REFERENCES "CARS" ("CAR\_ID") ON DELETE CASCADE ENABLE;

CREATE OR REPLACE EDITIONABLE TRIGGER "AUTOID\_ORDER"

before insert ON Orders\_c

for each row

declare

max\_n number;

pragma autonomous\_transaction;

begin

if :NEW.order\_id is null then

select max(order\_id) into max\_n from Orders\_c;

max\_n := max\_n + 1;

:NEW.order\_id := max\_n;

COMMIT;

end if;

end;

/

ALTER TRIGGER "AUTOID\_ORDER" ENABLE;

CREATE OR REPLACE EDITIONABLE TRIGGER "AUTO\_CONSULTING"

AFTER INSERT or Update ON Orders\_c

FOR EACH ROW

DECLARE

n\_id number;

no\_id number;

emdn\_id number;

sal number;

inc number;

ove number;

PRAGMA AUTONOMOUS\_TRANSACTION;

begin

Select SUM(orders\_c.order\_cost) into inc from orders\_c, orders\_consult where orders\_consult.status = 1 and orders\_c.order\_id = orders\_consult.order\_id;

inc := (inc \* 4) / 10;

Select SUM(EMPLOYEES\_C.emp\_salary) into sal from Employees\_c;

ove := inc - sal - 40000;

Insert into Income values(Sysdate, sal, inc, 40000, ove);

COMMIT;

Select max(order\_id) into n\_id from orders\_c;

Select max(order\_id) into no\_id from orders\_consult;

if n\_id != no\_id then

Select emp\_id into emdn\_id from EMPLOYEES\_C ORDER BY dbms\_random.value fetch first 1 row only;

Insert into Orders\_consult(order\_id, emp\_id, status) values (n\_id, emdn\_id, 0);

COMMIT;

end if;

end;

/

ALTER TRIGGER "AUTO\_CONSULTING" ENABLE;

**ORDERS\_CONSULT table:**

CREATE TABLE "ORDERS\_CONSULT"

( "ORDER\_ID" NUMBER,

"EMP\_ID" NUMBER,

"STATUS" NUMBER,

CONSTRAINT "ORDERS\_CONSULT\_CONPO" PRIMARY KEY ("ORDER\_ID")

USING INDEX ENABLE

) ;

ALTER TABLE "ORDERS\_CONSULT" ADD CONSTRAINT "ORDERS\_CONSULT\_CONFE" FOREIGN KEY ("EMP\_ID")

REFERENCES "EMPLOYEES\_C" ("EMP\_ID") ON DELETE CASCADE ENABLE;

ALTER TABLE "ORDERS\_CONSULT" ADD CONSTRAINT "ORDERS\_CONSULT\_CONFO" FOREIGN KEY ("ORDER\_ID")

REFERENCES "ORDERS\_C" ("ORDER\_ID") ON DELETE CASCADE ENABLE;

CREATE INDEX "ORDERS\_CONSULT\_IDXDWA" ON "ORDERS\_CONSULT" ("EMP\_ID")

;

CREATE OR REPLACE EDITIONABLE TRIGGER "ORDERS\_REPORT"

after insert or update or delete ON Orders\_consult

for each row

declare

PRAGMA AUTONOMOUS\_TRANSACTION;

all\_o number;

comp\_o number;

n\_comp\_o number;

begin

Select COUNT(order\_id) into all\_o from Orders\_consult;

Select COUNT(order\_id) into comp\_o from Orders\_consult where status = 1;

Select COUNT(order\_id) into n\_comp\_o from Orders\_consult where status = 0;

Insert into Orders\_status values(SYSDATE, all\_o, comp\_o, n\_comp\_o);

COMMIT;

end;

/

ALTER TRIGGER "ORDERS\_REPORT" ENABLE;

**ORDER\_STATUS table:**

CREATE TABLE "ORDERS\_STATUS"

( "UPDATE\_DATE" DATE,

"ORDERS\_NUM" NUMBER,

"COMPLETED" NUMBER,

"NOT\_COMPLETED" NUMBER,

CONSTRAINT "ORDERS\_STATUS\_CON" PRIMARY KEY ("UPDATE\_DATE")

USING INDEX ENABLE

) ;

**INCOME table:**

CREATE TABLE "INCOME"

( "UPDATE\_DATE" DATE,

"SALARY" NUMBER,

"SELL\_INCOME" NUMBER,

"LOSS" NUMBER,

"OVERALL" NUMBER,

CONSTRAINT "INCOME\_CONP" PRIMARY KEY ("UPDATE\_DATE")

USING INDEX ENABLE ) ;