

# **Database design & Modeling Assignment**

## **PART A: Coffee database**

- 1. Identify various functional dependencies. Using functional dependencies, evaluate the normal form of the table (Excel spreadsheet)? Justify.**

## **SOLUTION**

The attributes productID, areaCode and date are the composite primary keys of this table.

- ProductID, AreaCode -> Sales
- ProductID, AreaCode -> Inventory
- ProductID -> Product Name
- ProductID -> Product Line
- AreaCode -> Market Size

## **Reasons**

The data given in the table is in 1NF form since:

- The attributes have atomic values which are unbreakable.
- The data are redundant.
- Also some attributes like Product Name , Market Size etc are partially dependent on the composite primary keys (productID, areaCode and date).

- 2. Normalize the database into 3<sup>rd</sup> normal form using functional dependencies identified above. Identify the various tables including primary keys and foreign keys. (Feel free to create unique keys when you have composite primary keys)**

## **SOLUTION**

### **Product\_Details table**

<b>COLUMN</b>	<b>CONSTRAINT</b>
productID	Int,Primary key
product_Name	String,Unique key
prodLine	String
type	String
prodType	String

### **Product\_List table**

<b>COLUMN</b>	<b>CONSTRAINT</b>
productID	Int,Foreign Key
areacode	String,Foreign Key
Date	String,Primary Key
profit	Int
margin	Int
sales	Int
sdcogs	Int
totalExpenses	Int
marketing	Int
inventory	Int

budgetProfit	Int
budgetMargin	Int
budgetSales	Int
budgetCOGS	Int
product	Int

### **State\_Details table**

COLUMN	TYPE/CONSTRAINTS
stateID	Int,Primary Key,Unique key
state	Int
marketSize	Int

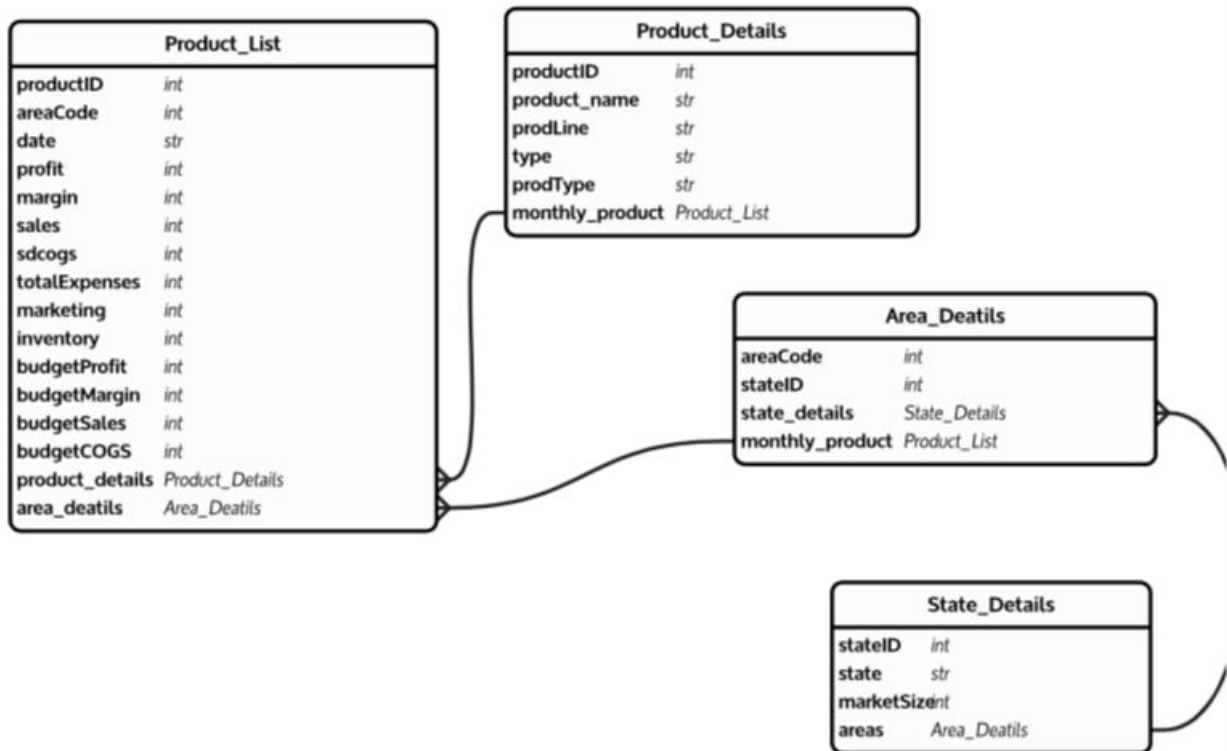
### **Area\_Details table**

COLUMN	TYPE/CONSTRAINTS
areaCode	Int,Primary Key,Unique key
state	Int
stateID	Int,Foreign Key

- Now, identify various entities and relationship types (1-1; 1-M; M-N), draw the E-R diagram using Oracle Data Modeler (or other tools). Convert the logical schema into actual table structure. Clearly identify PKs and FKs in the table structure.

## SOLUTION

### E-R Diagram:



### SQL :

```
CREATE TABLE "PRODUCT_DETAILS" (  
  "PRODUCTID" NUMBER(10) CONSTRAINT "PK_PRODUCT_DETAILS"  
  PRIMARY KEY,  
  "PRODUCT_NAME" VARCHAR2(1000 CHAR),  
  "PRODLINE" VARCHAR2(1000 CHAR),  
  "TYPE" VARCHAR2(1000 CHAR),  
  "PRODTYPE" VARCHAR2(1000 CHAR)  
);  
  
CREATE SEQUENCE "PRODUCT_DETAILS_SEQ" NOCACHE;  
  
CREATE TRIGGER "PRODUCT_DETAILS_BI"
```

```

BEFORE INSERT ON "PRODUCT_DETAILS"
FOR EACH ROW
BEGIN
    IF :new."PRODUCTID" IS NULL THEN
        SELECT "PRODUCT_DETAILS_SEQ".nextval INTO :new."PRODUCTID"
FROM DUAL;
    END IF;
END;;

```

```

CREATE TABLE "STATE_DETAILS" (
    "STATEID" NUMBER(10) CONSTRAINT "PK_STATE_DETAILS" PRIMARY
KEY,
    "STATE" VARCHAR2(1000 CHAR),
    "MARKETSIZE" NUMBER(10)
);

```

```

CREATE SEQUENCE "STATE_DETAILS_SEQ" NOCACHE;

```

```

CREATE TRIGGER "STATE_DETAILS_BI"
BEFORE INSERT ON "STATE_DETAILS"
FOR EACH ROW
BEGIN
    IF :new."STATEID" IS NULL THEN
        SELECT "STATE_DETAILS_SEQ".nextval INTO :new."STATEID" FROM
DUAL;
    END IF;
END;;

```

```

CREATE TABLE "AREA_DEATILS" (
    "AREACODE" NUMBER(10) CONSTRAINT "PK_AREA_DEATILS" PRIMARY
KEY,
    "STATEID" NUMBER(10),
    "STATE_DETAILS" NUMBER(10) NOT NULL
);

```

```

CREATE INDEX "IDX_AREA_DEATILS__STA_1f57e4ed" ON
"AREA_DEATILS" ("STATE_DETAILS");

```

```

ALTER TABLE "AREA_DEATILS" ADD CONSTRAINT
"FK_AREA_DEATILS__STATE_DETAILS" FOREIGN KEY
("STATE_DETAILS") REFERENCES "STATE_DETAILS" ("STATEID");

```

```
CREATE SEQUENCE "AREA_DEATILS_SEQ" NOCACHE;
```

```
CREATE TRIGGER "AREA_DEATILS_BI"  
  BEFORE INSERT ON "AREA_DEATILS"  
  FOR EACH ROW  
BEGIN  
  IF :new."AREACODE" IS NULL THEN  
    SELECT "AREA_DEATILS_SEQ".nextval INTO :new."AREACODE" FROM  
DUAL;  
  END IF;  
END;;
```

```
CREATE TABLE "PRODUCT_LIST" (  
  "PRODUCTID" NUMBER(10) CONSTRAINT "PK_PRODUCT_LIST" PRIMARY  
KEY,  
  "AREACODE" NUMBER(10),  
  "DATE" VARCHAR2(1000 CHAR),  
  "PROFIT" NUMBER(10),  
  "MARGIN" NUMBER(10),  
  "SALES" NUMBER(10),  
  "SDCOGS" NUMBER(10),  
  "TOTALEXPENSES" NUMBER(10),  
  "MARKETING" NUMBER(10),  
  "INVENTORY" NUMBER(10),  
  "BUDGETPROFIT" NUMBER(10),  
  "BUDGETMARGIN" NUMBER(10),  
  "BUDGETSALES" NUMBER(10),  
  "BUDGETCOGS" NUMBER(10),  
  "PRODUCT_DETAILS" NUMBER(10) NOT NULL,  
  "AREA_DEATILS" NUMBER(10) NOT NULL  
);
```

```
CREATE INDEX "IDX_PRODUCT_LIST__AREA_DEATILS" ON  
"PRODUCT_LIST" ("AREA_DEATILS");
```

```
CREATE INDEX "IDX_PRODUCT_LIST__PRO_59d2637f" ON  
"PRODUCT_LIST" ("PRODUCT_DETAILS");
```

```
ALTER TABLE "PRODUCT_LIST" ADD CONSTRAINT  
"FK_PRODUCT_LIST__AREA_DEATILS" FOREIGN KEY ("AREA_DEATILS")
```

```
REFERENCES "AREA_DEATILS" ("AREACODE");
```

```
ALTER TABLE "PRODUCT_LIST" ADD CONSTRAINT  
"FK_PRODUCT_LIST__PROD_1de5d347" FOREIGN KEY  
("PRODUCT_DETAILS") REFERENCES "PRODUCT_DETAILS"  
("PRODUCTID");
```

```
CREATE SEQUENCE "PRODUCT_LIST_SEQ" NOCACHE;
```

```
CREATE TRIGGER "PRODUCT_LIST_BI"  
BEFORE INSERT ON "PRODUCT_LIST"  
FOR EACH ROW  
BEGIN  
IF :new."PRODUCTID" IS NULL THEN  
SELECT "PRODUCT_LIST_SEQ".nextval INTO :new."PRODUCTID" FROM  
DUAL;  
END IF;  
END;
```

## **PART B:**

**Tommy and Tom (TT) Inc., a large manufacturer of electrical products located in Austin, TX, wants to build a database to track their parts, inventory, suppliers, and other data. Currently everything is stored in Excel spreadsheet. You are required to develop an E-R model before actually implementing a database.**

**TT sells hundreds of products and uses over 10,000 components in its products. The company knows precisely what parts are needed for each product. They maintain a list of suppliers who supply one or more of the components. Each supplier is evaluated on price/unit, quality, and reliability for each part it supplies (implies, a supplier can supply more than one part but the supplier is evaluated for each part separately). Each supplier is then ranked (that is, if there are 5 suppliers then each supplier will be ranked Number 1 to 5. One implies highest preference). The new system that the**

management is considering will require suppliers to update their lead-time, and minimum and maximum order quantity. For each part there is a re-order level. If the number of units falls below that re-order level then a new purchase order is released to a supplier based on the rank. The system maintains when an order was placed, expected delivery date, and actual delivery date. On receipt, parts are evaluated again for quality. A batch can be rejected if the quality is bad. If the parts supplied is certified “OK” then a supplier is a paid immediately through electronic money transfer.

**TASK:**

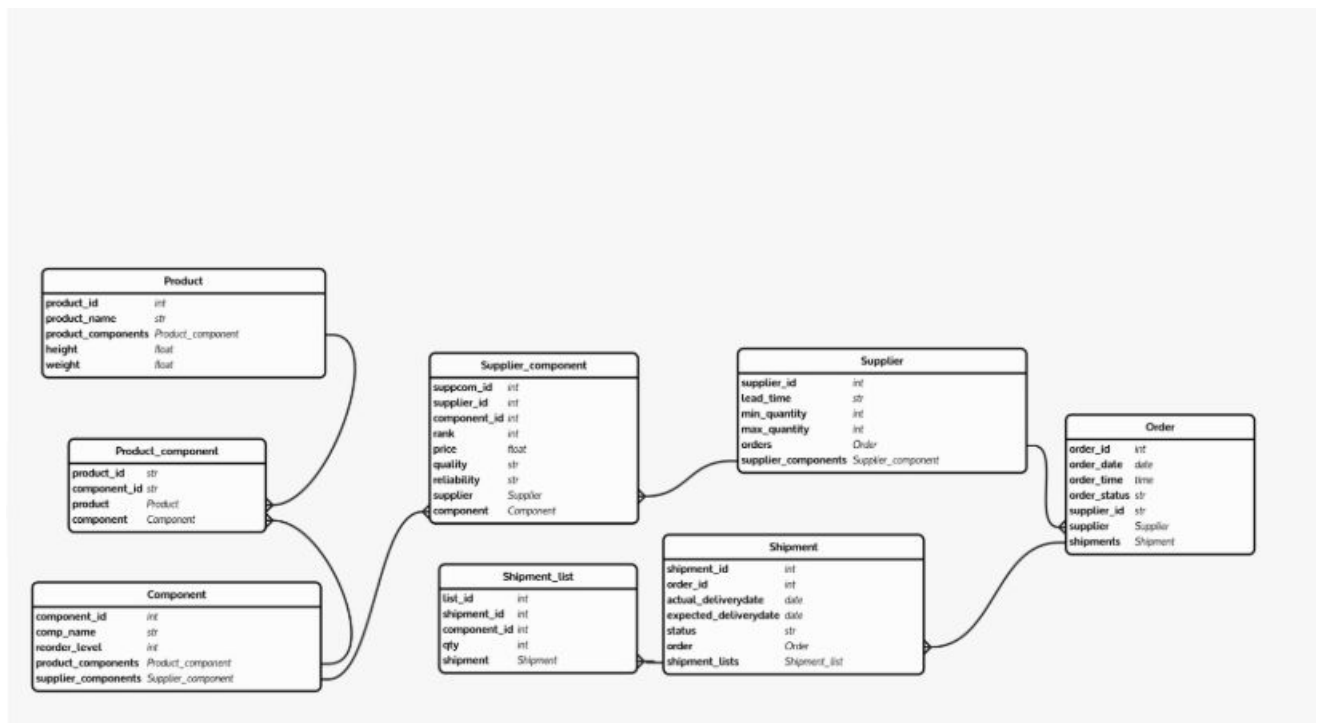
- a) Identify various entities and the relationships and draw the E-R diagram. Convert the E-R diagram into tables (relational model).
- b) What if an order can be supplied in small quantities at various times (that is, a large order can be split into many small orders). How does your E-R diagram and table structure change?

**SOLUTION**

To handle situation like “if an order can be supplied in small quantities at various times (that is, a large order can be split into many small orders)”.I have included another table to track shipment which have a have many-to-one relationship with the order table.This will help to track orders that are rejected,accepted etc.



## E-R Diagram :



## SQL :

```
CREATE TABLE "COMPONENT" (  
    "COMPONENT_ID" NUMBER(10) CONSTRAINT "PK_COMPONENT"  
    PRIMARY KEY,  
    "COMP_NAME" VARCHAR2(1000 CHAR) NOT NULL,  
    "REORDER_LEVEL" NUMBER(10) NOT NULL  
);
```

```
CREATE SEQUENCE "COMPONENT_SEQ" NOCACHE;
```

```
CREATE TRIGGER "COMPONENT_BI"  
  BEFORE INSERT ON "COMPONENT"  
  FOR EACH ROW  
BEGIN  
  IF :new."COMPONENT_ID" IS NULL THEN  
    SELECT "COMPONENT_SEQ".nextval INTO :new."COMPONENT_ID"  
  FROM DUAL;  
  END IF;  
END;;
```

```
CREATE TABLE "PRODUCT" (  
  "PRODUCT_ID" NUMBER(10) CONSTRAINT "PK_PRODUCT" PRIMARY  
KEY,  
  "PRODUCT_NAME" VARCHAR2(1000 CHAR) NOT NULL,  
  "HEIGHT" NUMBER,  
  "WEIGHT" NUMBER  
);
```

```
CREATE SEQUENCE "PRODUCT_SEQ" NOCACHE;
```

```
CREATE TRIGGER "PRODUCT_BI"  
  BEFORE INSERT ON "PRODUCT"  
  FOR EACH ROW  
BEGIN  
  IF :new."PRODUCT_ID" IS NULL THEN  
    SELECT "PRODUCT_SEQ".nextval INTO :new."PRODUCT_ID" FROM  
DUAL;  
  END IF;  
END;;
```

```
CREATE TABLE "PRODUCT_COMPONENT" (  
    "ID" NUMBER(10) CONSTRAINT "PK_PRODUCT_COMPONENT"  
PRIMARY KEY,  
    "PRODUCT_ID" VARCHAR2(1000 CHAR) NOT NULL,  
    "COMPONENT_ID" VARCHAR2(1000 CHAR) NOT NULL,  
    "PRODUCT" NUMBER(10) NOT NULL,  
    "COMPONENT" NUMBER(10) NOT NULL  
);
```

```
CREATE INDEX "IDX_PRODUCT_COMPONENT__PRODUCT" ON  
"PRODUCT_COMPONENT" ("PRODUCT");
```

```
CREATE INDEX "IDX_PRODUCT_COMPONENT_d560b912" ON  
"PRODUCT_COMPONENT" ("COMPONENT");
```

```
ALTER TABLE "PRODUCT_COMPONENT" ADD CONSTRAINT  
"FK_PRODUCT_COMPONENT__35db0c68" FOREIGN KEY  
("COMPONENT") REFERENCES "COMPONENT" ("COMPONENT_ID");
```

```
ALTER TABLE "PRODUCT_COMPONENT" ADD CONSTRAINT  
"FK_PRODUCT_COMPONENT__PRODUCT" FOREIGN KEY ("PRODUCT")  
REFERENCES "PRODUCT" ("PRODUCT_ID");
```

```
CREATE SEQUENCE "PRODUCT_COMPONENT_SEQ" NOCACHE;
```

```
CREATE TRIGGER "PRODUCT_COMPONENT_BI"  
BEFORE INSERT ON "PRODUCT_COMPONENT"  
FOR EACH ROW  
BEGIN  
    IF :new."ID" IS NULL THEN  
        SELECT "PRODUCT_COMPONENT_SEQ".nextval INTO :new."ID" FROM  
DUAL;  
    END IF;
```

END;;

```
CREATE TABLE "SUPPLIER" (  
    "SUPPLIER_ID" NUMBER(10) CONSTRAINT "PK_SUPPLIER" PRIMARY  
KEY,  
    "LEAD_TIME" VARCHAR2(1000 CHAR),  
    "MIN_QUANTITY" NUMBER(10),  
    "MAX_QUANTITY" NUMBER(10)  
);
```

```
CREATE SEQUENCE "SUPPLIER_SEQ" NOCACHE;
```

```
CREATE TRIGGER "SUPPLIER_BI"  
    BEFORE INSERT ON "SUPPLIER"  
    FOR EACH ROW  
BEGIN  
    IF :new."SUPPLIER_ID" IS NULL THEN  
        SELECT "SUPPLIER_SEQ".nextval INTO :new."SUPPLIER_ID" FROM  
DUAL;  
    END IF;  
END;;
```

```
CREATE TABLE "ORDER" (  
    "ORDER_ID" NUMBER(10) CONSTRAINT "PK_ORDER" PRIMARY KEY,  
    "ORDER_DATE" DATE NOT NULL,  
    "ORDER_TIME" INTERVAL DAY(0) TO SECOND(0) NOT NULL,  
    "ORDER_STATUS" VARCHAR2(1000 CHAR) NOT NULL,  
    "SUPPLIER_ID" VARCHAR2(1000 CHAR) NOT NULL,  
    "SUPPLIER" NUMBER(10) NOT NULL  
);
```

```
CREATE INDEX "IDX_ORDER__SUPPLIER" ON "ORDER" ("SUPPLIER");
```

```
ALTER TABLE "ORDER" ADD CONSTRAINT "FK_ORDER__SUPPLIER"  
FOREIGN KEY ("SUPPLIER") REFERENCES "SUPPLIER" ("SUPPLIER_ID");
```

```
CREATE SEQUENCE "ORDER_SEQ" NOCACHE;
```

```
CREATE TRIGGER "ORDER_BI"  
  BEFORE INSERT ON "ORDER"  
  FOR EACH ROW  
BEGIN  
  IF :new."ORDER_ID" IS NULL THEN  
    SELECT "ORDER_SEQ".nextval INTO :new."ORDER_ID" FROM DUAL;  
  END IF;  
END;;
```

```
CREATE TABLE "SHIPMENT" (  
  "SHIPMENT_ID" NUMBER(10) CONSTRAINT "PK_SHIPMENT" PRIMARY  
KEY,  
  "ORDER_ID" NUMBER(10) NOT NULL,  
  "ACTUAL_DELIVERYDATE" DATE,  
  "EXPECTED_DELIVERYDATE" DATE,  
  "STATUS" VARCHAR2(1000 CHAR) NOT NULL,  
  "ORDER" NUMBER(10) NOT NULL  
);
```

```
CREATE INDEX "IDX_SHIPMENT__ORDER" ON "SHIPMENT" ("ORDER");
```

```
ALTER      TABLE      "SHIPMENT"      ADD      CONSTRAINT  
"FK_SHIPMENT__ORDER" FOREIGN KEY ("ORDER") REFERENCES  
"ORDER" ("ORDER_ID");
```

```
CREATE SEQUENCE "SHIPMENT_SEQ" NOCACHE;
```

```
CREATE TRIGGER "SHIPMENT_BI"
```

```
BEFORE INSERT ON "SHIPMENT"
FOR EACH ROW
BEGIN
  IF :new."SHIPMENT_ID" IS NULL THEN
    SELECT "SHIPMENT_SEQ".nextval INTO :new."SHIPMENT_ID" FROM
DUAL;
  END IF;
END;;
```

```
CREATE TABLE "SHIPMENT_LIST" (
  "LIST_ID" NUMBER(10) CONSTRAINT "PK_SHIPMENT_LIST" PRIMARY
KEY,
  "SHIPMENT_ID" NUMBER(10) NOT NULL,
  "COMPONENT_ID" NUMBER(10),
  "QTY" NUMBER(10),
  "SHIPMENT" NUMBER(10) NOT NULL
);
```

```
CREATE      INDEX      "IDX_SHIPMENT_LIST__SHIPMENT"      ON
"SHIPMENT_LIST" ("SHIPMENT");
```

```
ALTER      TABLE      "SHIPMENT_LIST"      ADD      CONSTRAINT
"FK_SHIPMENT_LIST__SHIPMENT"      FOREIGN      KEY      ("SHIPMENT")
REFERENCES "SHIPMENT" ("SHIPMENT_ID");
```

```
CREATE SEQUENCE "SHIPMENT_LIST_SEQ" NOCACHE;
```

```
CREATE TRIGGER "SHIPMENT_LIST_BI"
  BEFORE INSERT ON "SHIPMENT_LIST"
  FOR EACH ROW
BEGIN
  IF :new."LIST_ID" IS NULL THEN
```

```
        SELECT "SHIPMENT_LIST_SEQ".nextval INTO :new."LIST_ID" FROM
DUAL;
    END IF;
END;;
```

```
CREATE TABLE "SUPPLIER_COMPONENT" (
        "SUPPCOM_ID"          NUMBER(10)          CONSTRAINT
"PK_SUPPLIER_COMPONENT" PRIMARY KEY,
    "SUPPLIER_ID" NUMBER(10) NOT NULL,
    "COMPONENT_ID" NUMBER(10) NOT NULL,
    "RANK" NUMBER(10) NOT NULL,
    "PRICE" NUMBER NOT NULL,
    "QUALITY" VARCHAR2(1000 CHAR) NOT NULL,
    "RELIABILITY" VARCHAR2(1000 CHAR) NOT NULL,
    "SUPPLIER" NUMBER(10) NOT NULL,
    "COMPONENT" NUMBER(10) NOT NULL
);
```

```
CREATE      INDEX      "IDX_SUPPLIER_COMPONENTEN_3943f517"      ON
"SUPPLIER_COMPONENT" ("COMPONENT");
```

```
CREATE      INDEX      "IDX_SUPPLIER_COMPONENTEN_5f991fe5"      ON
"SUPPLIER_COMPONENT" ("SUPPLIER");
```

```
ALTER TABLE "SUPPLIER_COMPONENT" ADD CONSTRAINT
"FK_SUPPLIER_COMPONENT_247042ca" FOREIGN KEY ("SUPPLIER")
REFERENCES "SUPPLIER" ("SUPPLIER_ID");
```

```
ALTER TABLE "SUPPLIER_COMPONENT" ADD CONSTRAINT
"FK_SUPPLIER_COMPONENT_ae97c99d" FOREIGN KEY ("COMPONENT")
REFERENCES "COMPONENT" ("COMPONENT_ID");
```

```
CREATE SEQUENCE "SUPPLIER_COMPONENT_SEQ" NOCACHE;
```

```
CREATE TRIGGER "SUPPLIER_COMPONENT_BI"  
  BEFORE INSERT ON "SUPPLIER_COMPONENT"  
  FOR EACH ROW  
BEGIN  
  IF :new."SUPPCOM_ID" IS NULL THEN  
    SELECT    "SUPPLIER_COMPONENT_SEQ".nextval    INTO  
:new."SUPPCOM_ID" FROM DUAL;  
  END IF;  
END;
```