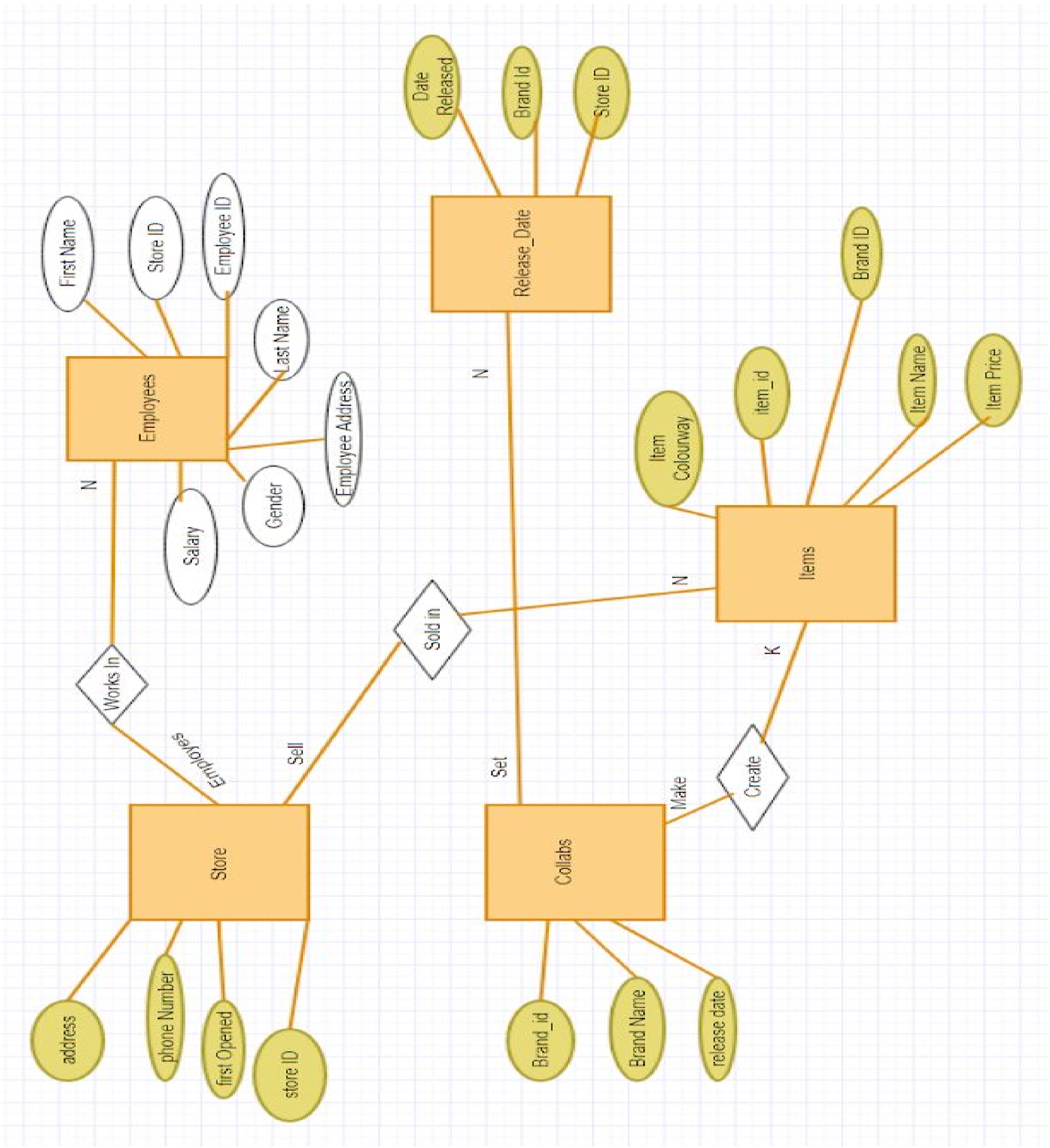
My Fashion Company Database(Supreme)

My streetwear company database is modeled after the brand Supreme.
There’s 2 views , the store view (this encompases employees) and their collaberations.

Entity Relationship Diagram



Store

The store table contains the store; id , name, address, phone number and the date it was 1st opened
The id of the stores is a primary key as each store id must be unique and it can’t be null ,i.e using NOT NULL.
All Supreme stores are just called Supreme , so the store name can’t be unique, however the different stores are differentiated by their location ,as the name=location >However each address is unique to that given store.(two stores don’t live under one address , or one store doesn’t have 2 addresses)

```
Create table stores(
Store_id NUMBER,
Store_name VARCHAR2(20) NOT NULL,
store_address VARCHAR2(300) UNIQUE,
Store_phone_number NUMBER,
Store_first_opened VARCHAR2(80 CHAR),
CONSTRAINT pk_stores PRIMARY KEY(Store_id)
);
```

(i) The store_id is a foreign key referencing the stor_id primary key , it won’t and can’t be null
(ii)There are 8 different employee tables for 8 of the current Supreme locations ; NY, LA,Brooklyn ,London,Paris,Japan Harajuku, Japan Shibuya and Japan Daikanyama.
They use Foreign keys to reference to the store table.Preventing the store_id in the Stores tabel from being null removes the chances of the corresponding employee table from being void. I do use DELETE CASCADE for the foreign key to to delete the corresponding records in the child table when the data in the parent table is deleted, in order to extra prevent there being employee data that doesn’t correspond to an existing store. (better to be on the safe side).

(iii) The CHECK constraint allows for only a specific set of valid values for the gender column (‘F’ - female, ‘M’ - male , ‘ND’ - not disclose)

```
CREATE TABLE employeesNY
(
Str_id NUMBER(1) NOT NULL, (i)
Employee_id VARCHAR2(20) NOT NULL,
first_name VARCHAR2(100),
last_name VARCHAR2(100),
gender VARCHAR(2) CONSTRAINT ck_gender CHECK (gender IN( ‘M’, ‘F’, ‘ND’ )), (iii)
Salary VARCHAR2(50) NOT NULL,
Employee_DOB DATE,
Employee_address VARCHAR2(100),
CONSTRAINT pk_employeesNY PRIMARY KEY (Employee_id),
CONSTRAINT fk_stores FOREIGN KEY (Str_id)
REFERENCES stores (Store_id) ON DELETE CASCADE   (ii)
);
```

In order to save time when creating the other tables for the employees for the other 7 locations , I used (below) to copy the columns of the employeesNY table without the data values.

```
CREATE TABLE employeesLon
AS SELECT*
FROM employeesNY
WHERE 3=5;
```

Triggers

I also included a trigger for when adding new employees to any of the given stores , that makes sure that an employee isn’t under 16. It does so by taking their D.O.B , subtracts it from the systems date and divides by 365.
(example below is for the the NY store)

```
CREATE OR REPLACE TRIGGER check_age
BEFORE INSERT OR UPDATE ON employeesNY
FOR EACH ROW
DECLARE
years_old NUMBER;
Error_msg VARCHAR(180);
BEGIN
years_old := ( (sysdate - : new.Employee_DOB) /365);
IF (years_old <16) THEN
Error_msg := ‘To young ,Do not hire’ ;
RAISE_APPLICATION_ERROR (-20601, error_msg);
END IF;
END;
```

Collabs

So Supreme has collaborated with many brands from Louis Vuitton, Nike,North Face etc
This view shows , the brands they’ve collaborated with , items sold, price , colours available , stores that they were sold at and their release dates.

In the main Collabs table , it classifies the Brand_id ,brand name , release date.
The Brand_id is a primary key and has the constraint of NOT NULL, for the same reason as the Stores_id.

There are 2 other tables in this view, release_date_location and item_released, that use the Collab_brand_id as a reference to their brand_id foreign keys.

```
CREATE TABLE collabs(
Collab_brand_id NUMBER NOT NULL,
Collab_brand_name VARCHAR2(300 CHAR),
Collab_brand_release_date DATE,
CONSTRAINT pk_collabs PRIMARY KEY(Collab_brand_id)
);
```

Release_date_location has attributes such as str_id , date released and brand_id.
There is a date released both in this table and the Collabs table as different stores release the collabs at different times (e.g Japan and online release items on later dates), so the release date in the Collabs table is more like the date the collab was announced and pictures of the pieces were made available.

```
CREATE TABLE release_date_location(
Brand_id NUMBER,
store_id NUMBER ,
Date_released DATE,
CONSTRAINT collabs_fk FOREIGN KEY (Brand_id) REFERENCES collabs
(Collab_brand_id) ON DELETE CASCADE,
CONSTRAINT stores_fk  FOREIGN KEY (store_id) REFERENCES stores(Store_id) ON
DELETE CASCADE
);
```

This table has 2 foreign keys referencing different tables , so I used the SELECT* FROM WHERE to save time and the data from those columns attributed to the ones for the release_date_location table entries.
I also used TO_DATE to have the date input and output in my preferred format .

```
INSERT INTO release_date_location VALUES((SELECT Collab_brand_id FROM collabs
WHERE Collab_brand_id = 1), (SELECT Store_id FROM stores WHERE Store_id = 1)
,TO_DATE(‘26/10/17’, 'DD/MM/YYYY') );
```

Items_released table has attributes such as brand_id, item_id, item name, item_price and item_colourways(as in the different colours the item comes in).

```
CREATE TABLE items_released(
Brand_id NUMBER,
Item_id NUMBER,
Item_name VARCHAR2(300 CHAR),
Item_price VARCHAR(30 CHAR),
Item_colourways VARCHAR(200 CHAR),
CONSTRAINT fk_collabs FOREIGN KEY (Brand_id) REFERENCES collabs
(Collab_brand_id) ON DELETE CASCADE
);
```

Similar to the the release_date_loction table , since the items_released table has a FOREIGN KEY that references attributes in another table , I used the SELECT.. FROM... WHERE … in order to save time and have the values from the referenced table (in this case it would be the collabs table , Collab_brand _id = 4 , which is the Scarface collab), correctly align with their released item(s).

```
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs
WHERE Collab_brand_id = 4),4.3, ‘The World is Yours Hooded Sweatshirt’ , ‘$188’,
‘detailed’);
```

Appendix - Commands

```
Create table stores(
        Store_id NUMBER,
        Store_name VARCHAR2(20) NOT NULL,
        store_address VARCHAR2(300) UNIQUE,
        Store_phone_number NUMBER,
        Store_first_opened VARCHAR2(80 CHAR),
        CONSTRAINT pk_stores PRIMARY KEY(Store_id)

);

INSERT INTO stores VALUES(1, ‘NewYork’ , ‘274 Lafayette Street’, 212-966-7799, ‘1994’);
INSERT INTO stores VALUES(2, ‘Brooklyn’, ‘152 Grand St. Brooklyn ,NY 11249’ , 718-599-2700
, ‘October 5 2017’ );
INSERT INTO stores VALUES(3, ‘Los Angeles’, ‘439 North Fairfax Ave, LA 90036’ ,
323-655-6205, ‘ 2004’);

INSERT INTO stores VALUES(4, ‘London’ , ‘2/3 Peter Street, London W1F 0AA’ ,
+44-207-437-0493 , ‘September 2011’);
INSERT INTO stores VALUES(5, ‘Paris’ , ‘20 Rue Barbette, Paris 75003’ , +33-1-43-48-80-14,
‘March 2016’ );
```

```
INSERT INTO stores VALUES(6, ‘Japan Shibuya‘ , ‘ 150-0041 Tokyo, Shibuya, 1-18-2 1F’
,+81-3-548-4394,’2012’);
INSERT INTO stores VALUES(7, ‘Japan Harajuku’ , ‘ 4 Chome-32-7 Jingumae’
,+81-3-5771-0090,’September 26th 2006’);
INSERT INTO stores VALUES(8, ‘Japan Daikanyama’ , ‘150-0034 Tokyo, Shibuya,
Daikanyamacho, 1−6’ , +81-3-5456-0085 ,’1998’);
INSERT INTO stores VALUES(9 , ‘Online’ ,’N/a ‘,0 , ‘N/A’);

CREATE TABLE employeesNY
(
        Str_id NUMBER(1) NOT NULL,
        Employee_id VARCHAR2(20) NOT NULL,
        first_name VARCHAR2(100),
        last_name VARCHAR2(100),
        gender VARCHAR(2) CONSTRAINT ck_gender CHECK (gender IN( ‘M’, ‘F’, ‘ND’ )),
        Salary VARCHAR2(50) NOT NULL,
        Employee_DOB DATE,
        Employee_address VARCHAR2(100),
        CONSTRAINT pk_employeesNY PRIMARY KEY (Employee_id),
        CONSTRAINT fk_stores FOREIGN KEY (Str_id)
        REFERENCES stores (Store_id) ON DELETE CASCADE

);
```

<u>Creating more tables by coping the columns of employeesNY, without the values</u>

```
CREATE TABLE employeesLon
AS SELECT*
FROM employeesNY
WHERE 3=5;

CREATE TABLE employeesLA
AS SELECT*
FROM employeesNY
WHERE 3=5;

CREATE TABLE employeesBrook
AS SELECT*
FROM employeesNY
WHERE 3=5;

CREATE TABLE employeesPR
AS SELECT*
FROM employeesNY
```

```
WHERE 3=5;

CREATE TABLE employeesTK_H
AS SELECT*
FROM employeesNY
WHERE 3=5;
CREATE TABLE employeesTK_Di
AS SELECT*
FROM employeesNY
WHERE 3=5;

CREATE TABLE employeesTK_S
AS SELECT*
FROM employeesNY
WHERE 3=5;
```

Triggers

```
CREATE OR REPLACE TRIGGER check_ageBrook
BEFORE INSERT OR UPDATE ON employeesBrook
FOR EACH ROW
DECLARE
years_old NUMBER;
Error_msg VARCHAR(180);
BEGIN
years_old := ( (sysdate - : new.Employee_DOB) /365);
IF (years_old <16) THEN
Error_msg := ‘To young ,Do not hire’ ;
RAISE_APPLICATION_ERROR (-20601, error_msg);
END IF;
END;


CREATE OR REPLACE TRIGGER check_ageLA
BEFORE INSERT OR UPDATE ON employeesLA
FOR EACH ROW
DECLARE
years_old NUMBER;
Error_msg VARCHAR(180);
BEGIN
years_old := ( (sysdate - : new.Employee_DOB) /365);
IF (years_old <16) THEN
Error_msg := ‘To young ,Do not hire’ ;
RAISE_APPLICATION_ERROR (-20601, error_msg);
END IF;
```

```
END;


CREATE OR REPLACE TRIGGER check_ageLon
BEFORE INSERT OR UPDATE ON employeesLon
FOR EACH ROW
DECLARE
years_old NUMBER;
Error_msg VARCHAR(180);
BEGIN
years_old := ( (sysdate - : new.Employee_DOB) /365);
IF (years_old <16) THEN
Error_msg := ‘To young ,Do not hire’ ;
RAISE_APPLICATION_ERROR (-20601, error_msg);
END IF;
END;


CREATE OR REPLACE TRIGGER check_ageTK_Di
BEFORE INSERT OR UPDATE ON employeesTK_Di
FOR EACH ROW
DECLARE
years_old NUMBER;
Error_msg VARCHAR(180);
BEGIN
years_old := ( (sysdate - : new.Employee_DOB) /365);
IF (years_old <16) THEN
Error_msg := ‘To young ,Do not hire’ ;
RAISE_APPLICATION_ERROR (-20601, error_msg);
END IF;
END;


CREATE OR REPLACE TRIGGER check_ageTK_H
BEFORE INSERT OR UPDATE ON employeesTK_H
FOR EACH ROW
DECLARE
years_old NUMBER;
Error_msg VARCHAR(180);
BEGIN
years_old := ( (sysdate - : new.Employee_DOB) /365);
IF (years_old <16) THEN
Error_msg := ‘To young ,Do not hire’ ;
RAISE_APPLICATION_ERROR (-20601, error_msg);
END IF;
END;
```

```
CREATE OR REPLACE TRIGGER check_ageTK_S
BEFORE INSERT OR UPDATE ON employeesTK_S
FOR EACH ROW
DECLARE
years_old NUMBER;
Error_msg VARCHAR(180);
BEGIN
years_old := ( (sysdate - : new.Employee_DOB) /365);
IF (years_old <16) THEN
Error_msg := ‘To young ,Do not hire’ ;
RAISE_APPLICATION_ERROR (-20601, error_msg);
END IF;
END;
```

```
CREATE OR REPLACE TRIGGER check_ageNY
BEFORE INSERT OR UPDATE ON employeesNY
FOR EACH ROW
DECLARE
years_old NUMBER;
Error_msg VARCHAR(180);
BEGIN
years_old := ( (sysdate - : new.Employee_DOB) /365);
IF (years_old <16) THEN
Error_msg := ‘To young ,Do not hire’ ;
RAISE_APPLICATION_ERROR (-20601, error_msg);
END IF;
END;
```

```
CREATE OR REPLACE TRIGGER check_agePR
BEFORE INSERT OR UPDATE ON employeesPR
FOR EACH ROW
DECLARE
years_old NUMBER;
Error_msg VARCHAR(180);
BEGIN
years_old := ( (sysdate - : new.Employee_DOB) /365);
IF (years_old <16) THEN
Error_msg := ‘To young ,Do not hire’ ;
RAISE_APPLICATION_ERROR (-20601, error_msg);
END IF;
END;
```

```
INSERT INTO employeesNY VALUES((SELECT Store_id FROM stores WHERE Store_id =
1),’1.1’,’Sarah’, ‘Dills’ , ‘F’, ‘$10 hr’ , TO_DATE(‘06/10/91’, 'DD/MM/YYYY'), ‘101 Private
Blue Lane’);
INSERT INTO employeesNY VALUES((SELECT Store_id FROM stores WHERE Store_id =
1),’1.2’,’Dylan’, ‘Sar’ , ‘M’ ,‘$10 hr’ , TO_DATE(‘26/01/89’, 'DD/MM/YYYY'), ‘101 Blue Lane,
P’);
INSERT INTO employeesNY VALUES((SELECT Store_id FROM stores WHERE Store_id =
1),’1.3’,’Shalom’, ‘Iss’ , ‘F’, ‘$12,50 hr’ , TO_DATE(‘17/06/94’, 'DD/MM/YYYY'), ‘10 P.Bue
Lane’);
INSERT INTO employeesNY VALUES((SELECT Store_id FROM stores WHERE Store_id =
1),’1.4’,’Mill’, ‘Dill’ , ‘M’, ‘$10 hr’ , TO_DATE(‘15/06/96’, 'DD/MM/YYYY'), ‘11 Priv Lane’);

INSERT INTO employeesNY VALUES((SELECT Store_id FROM stores WHERE Store_id =
1),’1.5’,’Cas’, ‘Mitch’ , ‘ND’ ‘$10 hr’ , TO_DATE(‘01/01/91’, 'DD/MM/YYYY'), ‘2 East Prite
BLane’);


INSERT INTO employeesLA VALUES((SELECT Store_id FROM stores WHERE Store_id =
2),’3.1’,’Grace’, ‘Fril’ , ‘F’, ‘$10 hr’ , TO_DATE(‘15/10/95’, 'DD/MM/YYYY'), ‘19 Bakers Blue
Lane’);
INSERT INTO employeesLA VALUES((SELECT Store_id FROM stores WHERE Store_id =
3),’3.2’,’Dral’, ‘Partl’ , ‘ND’ ,‘$10 hr’ , TO_DATE(‘16/12/90’, 'DD/MM/YYYY'), ‘25 B Bake
Plane’);
INSERT INTO employeesLA VALUES((SELECT Store_id FROM stores WHERE Store_id =
3),’3.3’,’Heela’, ‘Issa’ , ‘ND’, ‘$10,50 hr’ , TO_DATE(‘13/06/92’, 'DD/MM/YYYY'), ‘10 Pear.Bu
Port’);
INSERT INTO employeesLA VALUES((SELECT Store_id FROM stores WHERE Store_id =
3),’3.4’,’Mirth’, ‘Dray’ , ‘ND’, ‘$10 hr’ , TO_DATE(‘14/05/94’, 'DD/MM/YYYY'), ‘11 Pres Till
L’);
INSERT INTO employeesLA VALUES((SELECT Store_id FROM stores WHERE Store_id =
3),’3.5’,’Hesus’, ‘Meral’ , ‘ND’ ‘$10 hr’ , TO_DATE(‘02/12/91’, 'DD/MM/YYYY'), ‘2 EastBorn
Prite ’);


INSERT INTO employeesTK_S VALUES((SELECT Store_id FROM stores WHERE Store_id
= 6),’6.1’,’Nila’, ‘Rose’ , ‘F’, ‘$10 hr’ , TO_DATE(‘06/10/91’, 'DD/MM/YYYY'), ‘19A C-B K’);
INSERT INTO employeesTK_S VALUES((SELECT Store_id FROM stores WHERE Store_id
= 6),’6.2’,’Mar’, ‘Rior’ , ‘M’ ,‘$10 hr’ , TO_DATE(‘09/01/89’, 'DD/MM/YYYY'), ‘10X Hi Ti’);
INSERT INTO employeesTK_S VALUES((SELECT Store_id FROM stores WHERE Store_id
= 6),’6.3’,’Klar’, ‘Pa’ , ‘F’ ,‘$10 hr’ , TO_DATE(‘22/01/89’, 'DD/MM/YYYY'), ‘1 P Gean’);

INSERT INTO employeesTK_H VALUES((SELECT Store_id FROM stores WHERE Store_id
= 7),’7.1’,’Errl’, ‘Gull’ , ‘M’, ‘$10 hr’ , TO_DATE(‘17/07/93’, 'DD/MM/YYYY'), ‘70 FSA quad’);
INSERT INTO employeesTK_H VALUES((SELECT Store_id FROM stores WHERE Store_id
= 7),’7.2’,’Gra’, ‘Ren’ , ‘ND’ ,‘$10 hr’ , TO_DATE(‘19/09/94’, 'DD/MM/YYYY'), ‘99 Be L P’);
INSERT INTO employeesTK_H VALUES((SELECT Store_id FROM stores WHERE Store_id
= 7),’7.3’,’Dy’, ‘P’ , ‘M’ ,‘$10 hr’ , TO_DATE(‘13/03/93’, 'DD/MM/YYYY'), ‘33 T P’);
```

```
INSERT INTO employeesTK_Di VALUES((SELECT Store_id FROM stores WHERE
Store_id = 8),’8.1’, ‘Sahh’ , ‘Fi’,’ND’, ‘$10 hr’ , TO_DATE(‘06/06/96’, 'DD/MM/YYYY'), ‘66
Pate Lane’);
INSERT INTO employeesTK_Di VALUES((SELECT Store_id FROM stores WHERE
Store_id = 8),’8.2’, ‘Mal’ , ‘Sah’ ,’F’,‘$10 hr’ , TO_DATE(‘11/11/91’, 'DD/MM/YYYY'), ‘111 lue
LP’);
INSERT INTO employeesTK_Di VALUES((SELECT Store_id FROM stores WHERE
Store_id = 8),’8.3’,’Sar’, ‘De’ , ‘ND’ ,‘$10 hr’ , TO_DATE(‘22/02/92’, 'DD/MM/YYYY'), ‘22 BB
Lan’);
```

COLLABS

Collab table

```
CREATE TABLE collabs(
Collab_brand_id NUMBER NOT NULL,
Collab_brand_name VARCHAR2(300 CHAR),
Collab_brand_release_date DATE,
CONSTRAINT pk_collabs PRIMARY KEY(Collab_brand_id)
);

INSERT INTO collabs VALUES(1 ,’Nike Air Humara’ ,TO_DATE(
 ‘26/10/17’, 'DD/MM/YYYY') );
INSERT INTO collabs VALUES(2,’Nike SB’ , TO_DATE(‘07/09/17’,'DD/MM/YYYY'));
INSERT INTO collabs VALUES(3, ‘North Face’ ,TO_DATE( ‘19/10/17’,'DD/MM/YYYY'));
INSERT INTO collabs VALUES(4, ‘Scarface’, TO_DATE(‘12/09/17’,'DD/MM/YYYY'));
INSERT INTO collabs VALUES(5,’Doc Martens’ , TO_DATE(‘31/08/17’,'DD/MM/YYYY'));
```

Release date table

```
CREATE TABLE release_date_location(
Brand_id NUMBER,
store_id NUMBER ,
Date_released DATE,
CONSTRAINT collabs_fk FOREIGN KEY (Brand_id) REFERENCES collabs
(Collab_brand_id) ON DELETE CASCADE,
CONSTRAINT stores_fk  FOREIGN KEY (store_id) REFERENCES stores(Store_id) ON
DELETE CASCADE
);
INSERT INTO release_date_location VALUES((SELECT Collab_brand_id FROM collabs
WHERE Collab_brand_id = 1), (SELECT Store_id FROM stores WHERE Store_id = 1)
,TO_DATE(‘26/10/17’, 'DD/MM/YYYY') );
```

```
INSERT INTO release_date_location VALUES((SELECT Collab_brand_id FROM collabs
WHERE Collab_brand_id = 1), (SELECT Store_id FROM stores WHERE Store_id = 2)
,TO_DATE(‘26/10/17’, 'DD/MM/YYYY') );

INSERT INTO release_date_location VALUES((SELECT Collab_brand_id FROM collabs
WHERE Collab_brand_id = 1), (SELECT Store_id FROM stores WHERE Store_id = 3)
,TO_DATE(‘26/10/17’, 'DD/MM/YYYY') );

INSERT INTO release_date_location VALUES((SELECT Collab_brand_id FROM collabs
WHERE Collab_brand_id = 1), (SELECT Store_id FROM stores WHERE Store_id = 4)
,TO_DATE(‘26/10/17’, 'DD/MM/YYYY') );

INSERT INTO release_date_location VALUES((SELECT Collab_brand_id FROM collabs
WHERE Collab_brand_id = 1), (SELECT Store_id FROM stores WHERE Store_id = 5)
,TO_DATE(‘26/10/17’, 'DD/MM/YYYY') );

INSERT INTO release_date_location VALUES((SELECT Collab_brand_id FROM collabs
WHERE Collab_brand_id = 1), (SELECT Store_id FROM stores WHERE Store_id = 6)
,TO_DATE(‘09/09/17’, 'DD/MM/YYYY') );

INSERT INTO release_date_location VALUES((SELECT Collab_brand_id FROM collabs
WHERE Collab_brand_id = 1), (SELECT Store_id FROM stores WHERE Store_id = 7)
,TO_DATE(‘09/09/17’, 'DD/MM/YYYY') );

INSERT INTO release_date_location VALUES((SELECT Collab_brand_id FROM collabs
WHERE Collab_brand_id = 1), (SELECT Store_id FROM stores WHERE Store_id = 8)
,TO_DATE(‘09/09/17’, 'DD/MM/YYYY') );

INSERT INTO release_date_location VALUES((SELECT Collab_brand_id FROM collabs
WHERE Collab_brand_id = 1), (SELECT Store_id FROM stores WHERE Store_id = 9)
,TO_DATE(‘26/10/17’, 'DD/MM/YYYY') );
```

Items released table

```
CREATE TABLE items_released(
Brand_id NUMBER,
Item_id NUMBER,
Item_name VARCHAR2(300 CHAR),
Item_price VARCHAR(30 CHAR),
Item_colourways VARCHAR(200 CHAR),
CONSTRAINT fk_collabs FOREIGN KEY (Brand_id) REFERENCES collabs
(Collab_brand_id) ON DELETE CASCADE
);

INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 1), 1.3 , ‘ Air Humara’ , ‘$170’ , ‘black,eletric green ,electric blue,elctric magenta
with black detail’);
```

```
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 1), 1.1, ‘Trail Running Jacket’ , ‘$130’ , ‘black,eletric green ,electric blue,elctric
magenta with black detail’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 1), 1.2 , ‘Trail Running Pant’, ‘$90’ , ‘black,eletric green ,electric blue,elctric
magenta with black detail’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 1), 1.4, ‘ Trail Running Hat’ ,’$45’ ,’black,green ,blue,magenta with reflective
detailing’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 2), 2.1, ‘Air Force 2’ , ‘$98’ , ‘black,yellow,orange,light blue’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 3),3.1, ‘Leather Nuptse Jacket’ , ‘$1,098’ , ‘black,red,yellow’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 3),3.2, ‘Leather Base Camp Duffel’ ,’$388’,’black,red,yellow’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 3),3.3, ‘Leather Day Pack ‘, ‘$278’ ,’black,red,yellow’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 3),3.4,’Leather Roo II Lumber Pack’, ‘$148’ ,’black,red,yellow’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 3),3.5 , ‘Leather Gloves ‘ , ‘$148’ , ‘black,red,yellow’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 4),4.1, ‘Split T-Shirt’ , ‘$44’ , ‘red’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 4),4.2, ‘Blimp T-shirt’ , ‘$44’ , ‘black’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 4),4.3, ‘The World is Yours Hooded Sweatshirt’ , ‘$188’, ‘detailed’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 4),4.4,’The World Is Yours Lamp’ , ‘$168’, ‘detailed’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 4),4.4, ‘Split Skateboard’ , ‘$66‘, ‘detailed’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 4),4.5, ‘Scarface Sweater’ , ‘$178’ , ‘black/white’);
INSERT INTO items_released VALUES((SELECT Collab_brand_id FROM collabs WHERE
Collab_brand_id = 5),5.1, ‘3-Eye Shoe’ , ‘$168’, ‘dark red, black ,green’);
```