Patrick Page

5/1/2022

Databook: Part Three, The Database

Database Description

Stones Throw is an American independent record label that was founded in 1996 in Los Angeles, California. Currently, they are operating in a very archaic way with paper spreadsheets chaotically strewn about desktops in their Los Angeles office headquarters. Stones Throw founder and CEO, Peanut Butter Wolf, has recently contracted out work to DataPage, Inc. for the creation of a database that will keep tabs on almost every facet of his record label business.

Stones Throw has been operational for more than 25 years and has already published an expansive catalog of music that consists of work from hundreds of different artists. As years and years go by, the label continues to grow. Peanut Butter Wolf is hiring more staff and interns than ever before. The necessity for a database is evident when trying to keep track of all the employees, artists, producers, tour dates, and discographies.

Additional to the music sales, Stones Throw also has a thriving online shop where each artist on the label can sell their own merchandise. Peanut Butter Wolf would like the database to help process all of the sales and transactions that take place on the label's website.

The different manufacturers and distributors that the label works with are crucial to the business for creating the merchandise and pressing the vinyl records as well as shipping the items anywhere around the globe. For this reason, it is another business requirement for the database to keep critical information about all of the different manufacturing and distributing companies Stones Throw works with.

Primary users of the database will be made up of different groups of employees within the label. Each primary user group will require the database to contain specific sets of information that will help them effectively do their daily jobs.

The **executives** of Stones Throw require the database to reference employee details like name, address, contact information, job title, and salary. Historical information about all of the past and current albums would also be required for archiving purposes. They also need to know how well the records are selling and what profits are being made by both the merchandise and music sales.

Employees that make up the **artist and repertoire (A&R) team** need to know what artists are currently signed and what artists have been scouted. They need to know who the agents are for the different artists and producers that are affiliated with the label.

The **sales team** needs to know how well each album released by the label has done on the market. They need to know which distributors are currently responsible for distributing records and merchandise from the different suppliers. They need to coordinate with the manufacturers to understand when to ask for new shipments of merchandise or vinyl, and they also need to know inventory and sales information of all the merchandise on the online store so that transactions can be processed.

The **legal team** needs to keep track of the royalties the record label is responsible for honoring. These royalties will most likely occur due to songs released by Stones Throw using licensed samples from songs released by other artists on other record labels or platforms. The legal team needs to understand what royalties already exist and add new royalties as they occur so that the correct percentages of profits are being distributed appropriately.

Lastly, the **agents/manager** of the musical artists need to know the touring schedule for the artist that they manage. Stones Throw is planning on creating a new UI that will help aid the artists with scheduling while on these tours.

Data Dictionary

TABLE NAME	ATTRIBUTE NAME	CONTENTS	TYPE	FORMAT	RANGE	REQUIRED	PK OR FK	FK REFERENCED TABLE
EMPLOYEES	EMP_ID	Employee code	INTEGER	####	NA	Y	PK	
	EMP_FNAME	Employee first name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	EMP_LNAME	Employee last name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	EMP_BDAY	Employee birth date	DATE	MM-DD-YYYY	NA	Y		
	EMP_ADDRESS	Emplyee address	VARCHAR(35)	Xxxxxxxx	NA	Y		
	EMP_CITY	Employee city	VARCHAR(35)	Xxxxxxxx	NA	Y		
	EMP_STATE	Employee state	CHAR(2)	XX	NA	Y		
	EMP_PHONE	Employee phone #	CHAR(12)	999-999-9999	NA	Y		
	EMP_EMAIL	Employee email	VARCHAR(35)	Xxxxxxxx	NA	Y		
	EMP_TITLE	Employee job title	VARCHAR(35)	Xxxxxxxx	NA	Y		
	EMP_HIRE	Employee hire date	DATE	MM-DD-YYYY	NA	Y		
	EMP_SALARY	Employee salary	NUMERIC(8,2)	######.##	25000.00 - 500000.00	Y		
ARTISTS	ART_ID	Artist code	INTEGER	####	NA	Y	PK	
	ART_STAGE	Artist stage name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	ART_FNAME	Artist first name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	ART_LNAME	Artist last name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	ART_MANAGER	Artist's manager	VARCHAR(35)	Xxxxxxxx	NA	Y		
PRODUCERS	PROD_ID	Producer code	INTEGER	####	NA	Y	PK	
	PROD_STAGE	Producer stage name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	PROD_FNAME	Producer first name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	PROD_LNAME	Producer last name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	PROD_MANAGER	Producer's manager	VARCHAR(35)	Xxxxxxxx	NA			
DISTRIBUTORS	DIST_ID	Distributor code	INTEGER	####	NA	Y	PK	
	DIST_NAME	Distributor name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	DIST_ADDRESS	Distributor address	VARCHAR(35)	Xxxxxxxx	NA	Y		
	DIST_CITY	Distributor city	VARCHAR(35)	Xxxxxxxx	NA	Y		
	DIST_STATE	Distributor state	CHAR(2)	XX	NA	Y		
TRACKS	TRACK_ID	Track code	INTEGER	####	NA	Y	PK	
	TRACK_NAME	Track name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	DISC_ID	Track album code	INTEGER	####	NA	Y	FK	DISCOGRAPHY
	PROD_ID	Track producer	INTEGER	####	NA	Y	FK	PRODUCERS
	ROYALTY_ID	Track royalty code	INTEGER	####	NA		FK	ROYALTIES
DISCOGRAPHY	DISC_ID	Discography code	INTEGER	####	NA	Y	PK	
	DISC_ALBUM	Album name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	ARTIST_ID	Album artist	INTEGER	####	NA	Y	FK	ARTISTS
	DISC_RELEASE	Album release date	DATE	MM-DD-YYYY	NA	Y		
	DISC_SALES	Album sales	NUMERIC(12,2)	##########.##	0.00 - 999999999.99	Y		
	DISC_PROFIT	Album profit	NUMERIC(12,2)	##########.##	0.00 - 999999999.99	Y		
	DIST_ID	Album distributor code	INTEGER	####	NA	Y	FK	DISTRIBUTORS
SALES	SALES_ID	Sales code	INTEGER	####	NA	Y	PK	
	MERCH_ID	Merchandise code	INTEGER	####	NA	Y	FK	MERCHANDISE
	SALES_BUYER	Sales item purchaser	VARCHAR(35)	Xxxxxxxx	NA	Y		
	SALES_XSHIP	Sales expedited shipping (Y/N)	BIT	0 or 1	NA	Y		
	SALES_ADDRESS	Sales shipping address	VARCHAR(35)	Xxxxxxxx	NA	Y		
ROYALTIES	ROYALTY_ID	Royalty code	INTEGER	####	NA	Y	PK	
	ROYALTY_ARTIST	Royalty artist	VARCHAR(35)	Xxxxxxxx	NA	Y		
	ROYALTY_PERCENTAGE	Royalty percentage	DECIMAL(3,3)	0.###	0.000 - 0.999	Y		
MANUFACTURERS	MAN_ID	Manufacturer code	INTEGER	####	NA	Y	PK	
	MAN_NAME	Manufacturer name	VARCHAR(35)	Xxxxxxxx	NA	Y		
	MAN_ADDRESS	Manufacturer address	VARCHAR(35)	Xxxxxxxx	NA	Y		
	MAN_CITY	Manufacturer city	VARCHAR(35)	Xxxxxxxx	NA	Y		
	MAN_STATE	Manufacturer state	CHAR(2)	XX	NA	Y		
	MAN_REP	Manufacturer rep	VARCHAR(35)	Xxxxxxxx	NA	Y		
	MAN_CONTACT	Manufacturer contact	VARCHAR(35)	Xxxxxxxx	NA	Y		
MERCHANDISE	MERCH_ID	Merchandise code	INTEGER	####	NA	Y	PK	
	MERCH_STOCK	Merchandise stock	INTEGER	####	0 - 9999	Y		
	ART_ID	Merchandise artist	INTEGER	####	NA	Y	FK	ARTISTS
	MERCH_ITEM	Merchandise item type	VARCHAR(35)	Xxxxxxxx	NA	Y		
	MERCH_COST	Merchandise cost	NUMERIC(6,2)	####.##	0 - 999.99	Y		
	MAN_ID	Merchandise manufacturer code	INTEGER	####	NA	Y	FK	MANUFACTURERS
	DIST_ID	Merchandise distributor code	INTEGER	####	NA	Y	FK	DISTRIBUTORS
TOURING	TOUR_ID	Tour code	INTEGER	####	NA	Y	PK	
	ARTIST_ID	Tour artist code	INTEGER	####	NA	Y	FK	ARTISTS
	TOUR_CONTINENT	Tour continent	VARCHAR(35)	Xxxxxxxx	NA	Y		
	TOUR_SHOWS	Tour number of shows	INTEGER	####	0 - 99	Y		
	TOUR_GUEST	Tour guest	VARCHAR(35)	Xxxxxxxx	NA			

Business Rules

1. One song is contained on one album. (1:M)
2. One album contains many songs.
3. One song is produced by one or many producer(s). (M:N)
4. One producer can produce many songs.
5. One song owes royalties to one or many artist(s). (M:N)
6. One artist can be owed many royalties.
7. One album is made by one artist. (1:M)
8. One artist can make many albums.
9. One main artist goes on one tour. (1:1)
10. One tour is composed of one main artist.
11. One merchandise item is contained in one sale. (1:M)
12. One sale can contain many merchandise items.
13. One merchandise item is made by one manufacturer. (1:M)
14. One manufacturer makes many merchandise items.
15. One album is distributed by many distributors. (M:N)
16. One distributor can distribute many albums.
17. One merchandise item is distributed by many distributors. (M:N)
18. One distributor can distribute many merchandise items.

Entity Relationship Model Components

ENTITY	RELATIONSHIP	CONNECTIVITY	ENTITY
ALBUM	has	1:M	TRACKS
PRODUCERS	produces	M:N	TRACKS
TRACKS	have	M:N	ROYALTIES
ARTIST	makes	1:M	DISCOGRAPHY
ARTIST	goes on	1:1	TOUR
SALES	contain	1:M	MERCHANDISE
MANUFACTURER	makes	1:M	MERCHANDISE
DISTRIBUTOR	distributes	M:N	DISCOGRAPHY
DISTRIBUTOR	distributes	M:N	MERCHANDISE

Shorthand Relational Schemas

EMPLOYEES (**EMP_ID**, EMP_FNAME, EMP_LNAME, EMP_BDAY, EMP_ADDRESS, EMP_CITY, EMP_STATE, EMP_PHONE, EMP_EMAIL, EMP_TITLE, EMP_HIRE, EMP_SALARY)

ARTISTS **(ART_ID**, ART_STAGE, ART_FNAME, ART_LNAME, ART_MANAGER)

PRODUCERS (**PROD_ID**, PROD_STAGE, PROD_FNAME, PROD_LNAME, PROD_MANAGER)

DISTRIBUTORS (**DIST_ID**, DIST_NAME, DIST_ADDRESS, DIST_CITY, DIST_STATE)

TRACKS (**TRACK_ID**, DISC_ID, PROD_ID, ROYALTY_ID, TRACK_NAME)

DISCOGRAPHY (**DISC_ID**, ART_ID, DIST_ID, DISC_ALBUM, DISC_RELEASE, DISC_SALES, DISC_PROFIT)

SALES (**SALES_ID**, MERCH_ID, SALES_BUYER, SALES_XSHIP, SALES_ADDRESS)

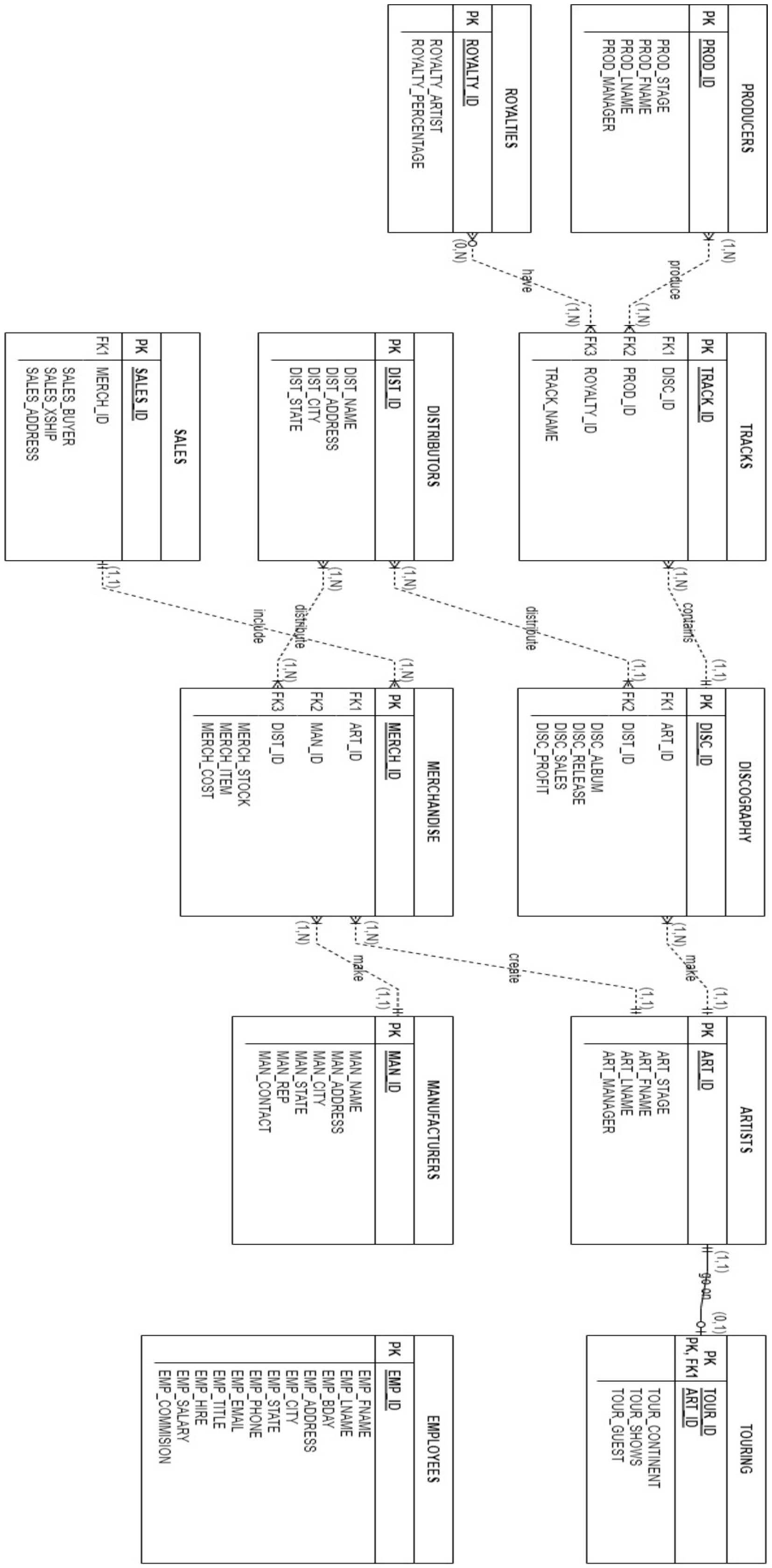
ROYALTIES (**ROYALTY_ID**, ROYALTY_ARTIST, ROYALTY_PERCENTAGE)

MANUFACTURERS (**MAN_ID**, MAN_NAME, MAN_ADDRESS, MAN_CITY, MAN_STATE, MAN_REP, MAN_CONTACT)

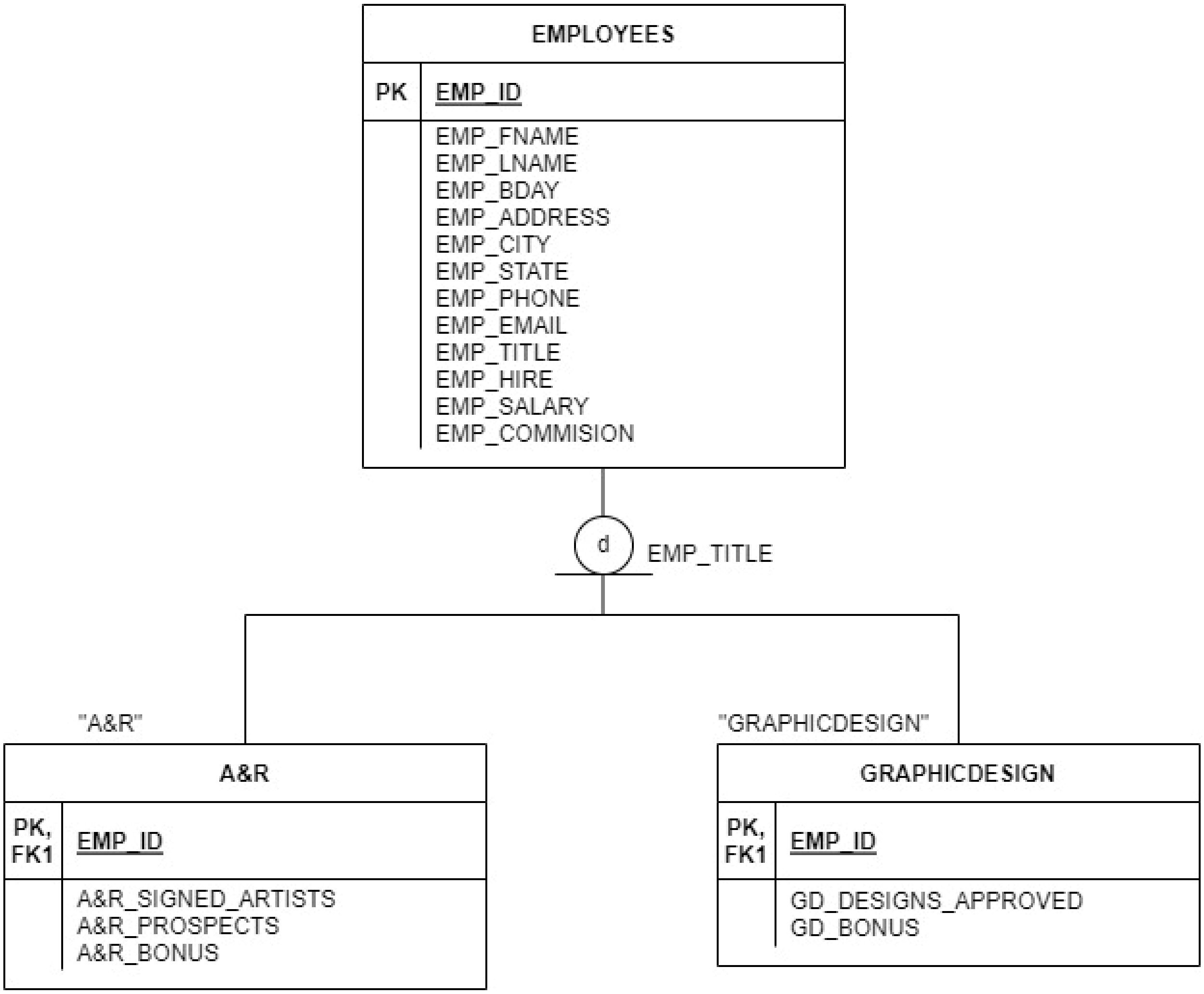
MERCHANDISE (**MERCH_ID**, ART_ID, MAN_ID, DIST_ID, MERCH_STOCK, MERCH_ITEM, MERCH_COST)

TOURING (**TOUR_ID**, **ART_ID**, TOUR_CONTINENT, TOUR_SHOWS, TOUR_GUEST)

ERD


PRODUCERS
PK PROD_ID
PROD_STAGE
PROD_FNAME
PROD_LNAME
PROD_MANAGER
ROYALTIES
PK ROYALTY_ID
ROYALTY_ARTIST
ROYALTY_PERCENTAGE
(1,N)
produce
(1,N)
(0,N)
have
(1,N)
TRACKS
PK TRACK_ID
FK1 DISC_ID
FK2 PROD_ID
FK3 ROYALTY_ID
TRACK_NAME
DISTRIBUTORS
PK DIST_ID
DIST_NAME
DIST_ADDRESS
DIST_CITY
DIST_STATE
SALES
PK SALES_ID
FK1 MERCH_ID
SALES_BUYER
SALES_XSHIP
SALES_ADDRESS
(1,N)
contains
(1,1)
(1,N)
distribute
(1,1)
(1,N)
distribute
(1,N)
(1,1)
include
(1,N)
DISCOGRAPHY
PK DISC_ID
FK1 ART_ID
FK2 DIST_ID
DISC_ALBUM
DISC_RELEASE
DISC_SALES
DISC_PROFIT
MERCHANDISE
PK MERCH_ID
FK1 ART_ID
FK2 MAN_ID
FK3 DIST_ID
MERCH_STOCK
MERCH_ITEM
MERCH_COST
(1,N)
make
(1,1)
(1,N)
create
(1,1)
(1,N)
make
(1,1)
ARTISTS
PK ART_ID
ART_STAGE
ART_FNAME
ART_LNAME
ART_MANAGER
MANUFACTURERS
PK MAN_ID
MAN_NAME
MAN_ADDRESS
MAN_CITY
MAN_STATE
MAN_REP
MAN_CONTACT
(1,1)
go on
(0,1)
TOURING
PK TOUR_ID
PK, FK1 ART_ID
TOUR_CONTINENT
TOUR_SHOWS
TOUR_GUEST
EMPLOYEES
PK EMP_ID
EMP_FNAME
EMP_LNAME
EMP_BDAY
EMP_ADDRESS
EMP_CITY
EMP_STATE
EMP_PHONE
EMP_EMAIL
EMP_HIRE
EMP_TITLE
EMP_SALARY
EMP_COMMISION

Subtype/Supertype Diagram

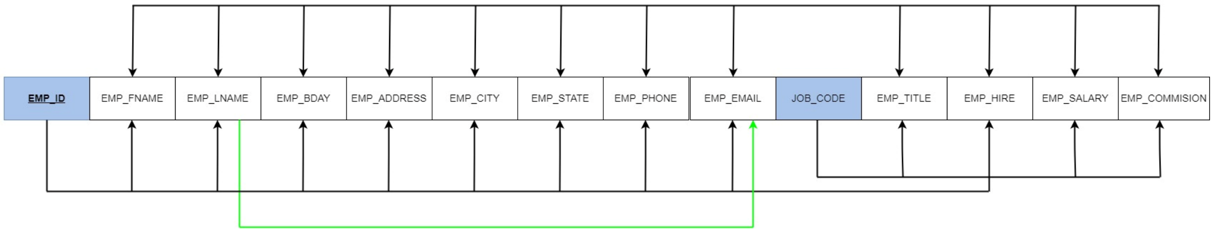


Normalization of EMPLOYEES

1NF (**EMP_ID**, EMP_FNAME, EMP_LNAME, EMP_BDAY, EMP_ADDRESS, EMP_CITY, EMP_STATE, EMP_PHONE, EMP_EMAIL, **JOB_CODE,** EMP_TITLE, EMP_HIRE, EMP_SALARY, EMP_COMMISION)

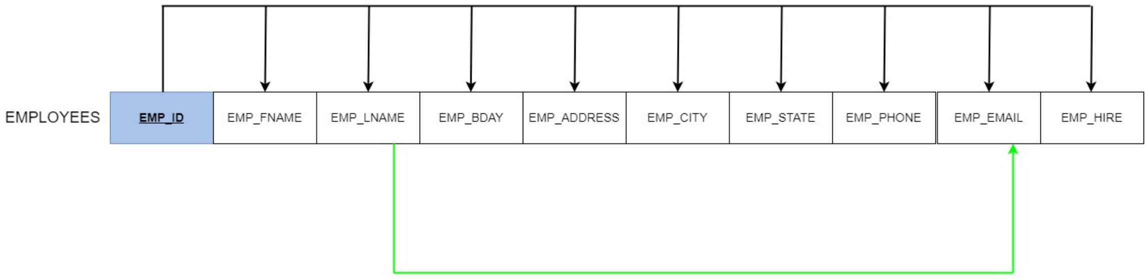
Partial dependencies: (**EMP_ID** —> EMP_FNAME, EMP_LNAME, EMP_BDAY, EMP_ADDRESS, EMP_CITY, EMP_STATE, EMP_PHONE, EMP_EMAIL, EMP_TITLE, EMP_HIRE, EMP_SALARY, EMP_COMMISION), (**JOB_CODE** —> EMP_TITLE, EMP_SALARY, EMP_COMMISION)

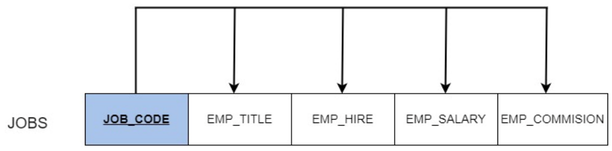
Transitive dependencies: (EMP_LNAME —> EMP_EMAIL)



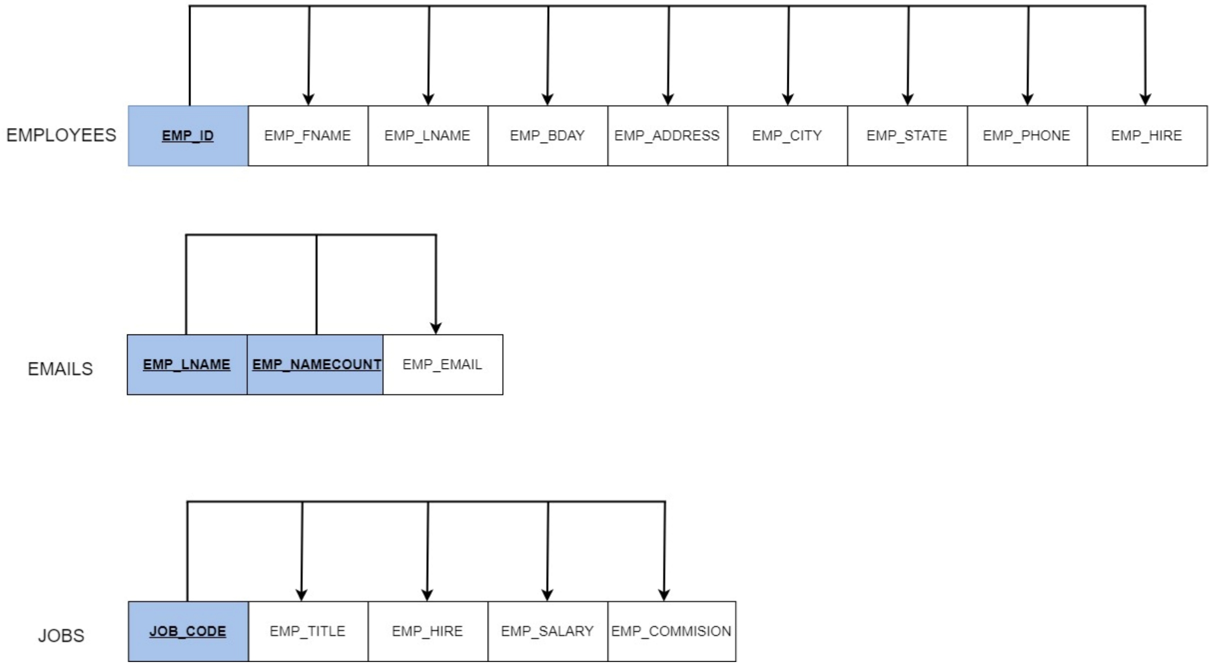
2NF EMPLOYEES (**EMP_ID**, EMP_FNAME, EMP_LNAME, EMP_BDAY, EMP_ADDRESS, EMP_CITY, EMP_STATE, EMP_PHONE, EMP_EMAIL, **JOB_CODE,** EMP_HIRE), JOBS (**JOB_CODE,** EMP_TITLE, EMP_SALARY, EMP_COMMISION)

Transitive dependencies: (EMP_LNAME —> EMP_EMAIL)





3NF EMPLOYEES (**EMP_ID**, EMP_FNAME, EMP_LNAME, EMP_BDAY, EMP_ADDRESS, EMP_CITY, EMP_STATE, EMP_PHONE, **JOB_CODE,** EMP_HIRE), JOBS (**JOB_CODE,** EMP_TITLE, EMP_SALARY, EMP_COMMISION), EMAILS (**EMP_LNAME**, **EMP_NAMECOUNT**, EMP_EMAIL)



The 'EMAILS' table will automatically generate an email based on the employees' last name. For instance, the employee John Doe will be assigned the email doe@stones-throw.com. If there are two employees with the last name "Doe", then the most recent entry will be given an EMP_NAMECOUNT of "2" therefor the email assigned will be doe2@stones-throw.com.

Questions and Queries

3 Questions:

1. Who has been employed at Stones Throw the longest?
2. How many tracks has Madlib produced on the Stones Throw label?
3. How many merchandise items are being sold for artists that are actively touring?

3 Queries:

1.
   ```
   SELECT EMP_FNAME, EMP_LNAME, EMP_HIRE FROM `EMPLOYEES` ORDER BY EMP_HIRE ASC LIMIT 1;
   ```

2.
   ```
   SELECT PROD_STAGE, COUNT(TRACK_ID) FROM `TRACKS` JOIN `PRODUCERS` ON PRODUCERS.PROD_ID = TRACKS.PROD_ID WHERE PROD_STAGE like 'Madlib';
   ```

3.
   ```
   SELECT ART_ID, COUNT(MERCH_ID) FROM `MERCHANDISE` WHERE ART_ID IN (SELECT DISTINCT ART_ID FROM `TOURING`) GROUP BY ART_ID;
   ```