Part 1: Database Schema Design

E-R Gram below:

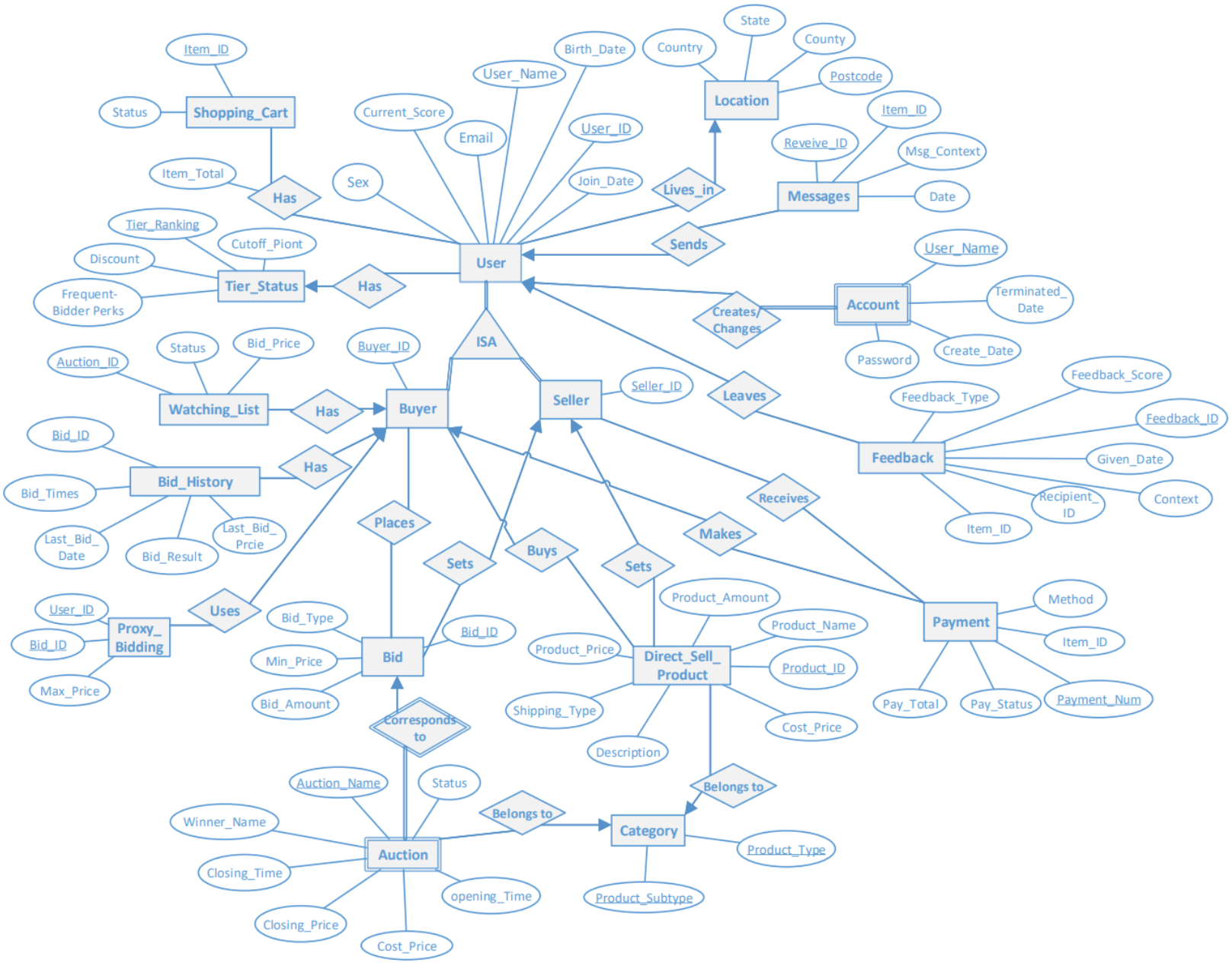

Table Representations:

User – Shopping Cart

User

User_ID	User_Name	Sex	Birth_Date	Email	Join_Date	Current_Score	Current_Rank	Postcode
0001	jjcool	F	1999/08/20	jjxx@jhu.edu	2018/01/01	200	Gold	21218
0002	IlikePonies	M	1982/02/14	cc@gmail.com	2008/04/15	134	Silver	21101
0003	MileyC	F	1992/06/17	mileyc@jhu.edu	2017/02/10	350	ShiningStar	21218
0004	ICantStop	F	2000/01/08	stt@jj.com	2020/08/11	400	ShiningStar	20124

Shopping Cart

Item_ID	Status
b-01	unsold
s-03	unsold

User's Cart

Item_ID	User_ID	Item_Total
b-01	0001	1
s-04	0002	2

User – Location

Location

Postcode	State	County	Country
21214	Maryland	Towson	United States
21101	Maryland	Owings Mill	Uniterd States

User – Messages

Messages

Receive_ID	Item_ID	User_ID	Msg_Context	Date
0023	b-01	0001	Great!	2021/05/20
0056	b-02	0001	Not bad!	2020/08/12

User – Tier Rankings

Tier-Status

Tier_Ranking	Cutoff_Point	Discount	Frequent-Bidder Perks
Gold	200	10%	20$
Silver	100	15%	30$

User – Account (Username cannot be the same, except being terminated)

Account

User_Name	UserID	Create_Date	Terminated_Date	User_ID	Password
ImTooShy	0001	2018/01/01	2019/02/10	0001	123456
ImTooTimid	0001	2019/02/10	Null	0001	123456
ImTooShy	0002	2019/05/18	Null	0048	654321

User – Buyer Seller

Buyer

Buyer_ID	User_Name	Sex	Birth_Date	Email	Join_Date	Current_Score	Current_Rank	Postcode
0001	jjcool	F	1999/08/20	jjxx@jhu.edu	2018/01/01	200	Gold	21218
0002	IlikePonies	M	1982/02/14	cc@gmail.com	2008/04/15	400	Ruby	21101
0003	MileyC	F	1992/06/17	mileyc@jhu.edu	2017/02/10	500	Gold	21218
0004	ICantStop	F	2000/01/08	stt@jj.com	2020/08/11	100	Ruby	20124

Seller

Seller_ID	User_Name	Sex	Birth_Date	Email	Join_Date	Current_Score	Current_Rank	Postcode
0002	IlikePonies	M	1982/02/14	cc@gmail.com	2008/04/15	400	Ruby	21101

User – Feedback

Feedback

Feedback_ID	Feedback_Type	Feedback_Score	Context	User_ID	Recipient_ID	Item_ID	Given_Date
f-01	Positive	10	Great	0001	0023	b-01	2021/10/02
f-02	Bad	-10	Awful	0032	0045	b-03	2021/09/10

Buyer – payment

Payment

Payment_Number	**Item_ID**	**Buyer_ID**	**Method**	**Pay_Total**	**Pay_Status**
p-01	b-01	0001	Paypal	50$	Success
p-02	b-05	0014	Credit Card	65$	Success

Seller Receive Payment

Receive_Payment

Payment_Number	**Seller-ID**
p-01	0023
p-02	0053

Buyer – Watching list

Watching_List

Bid_ID	**Buyer_ID**	**Status**	**Bid_Price**
b-05	0001	Unsold	150$
b-27	0001	Sold	200$

Buyer – Bid_History

Bid_History

Bid_ID	**Buyer_ID**	**Bid_Times**	**Last_Bid_Date**	**Bid_Result**	**Last_Bid_Price**
b-01	0001	2	09-28-2021	Success	50$
b-03	0004	1	10-11-2020	Success	85$
b-03	0003	4	09-20-2021	Failed	60$
b-05	0004	3	05-14-2021	Success	65$
b-10	0041	1	12-10-2020	Declined	100$

Buyer – Proxy bidding

Proxy_Bidding

Bid_ID	**Buyer_ID**	**Max_Price**
b-01	0001	60$
b-02	0001	45$
b-05	0004	80$
b-03	0004	40$

Buyer – Bid

Bid

Bid_ID	**Seller_ID**	**Bid_Type**	**Min_Price**	**Bid_Amount**
b-01	0002	Standard	20$	1
b-03	0002	Dutch	30$	10
b-05	0013	Standard	50$	1

Place_Bid

Buyer_ID	**Bid_ID**
0001	b-01

0004	b-03

Bid – Auction

Auction

Bid_ID	**Auction_Name**	**Status**	**Winner_Name**	**Opening_Time**	**Closing_Time**	**Closing_Price**	**Cost_Price**
b-01	Babyplus	Sold	jjcool	2021/09/20	2021/09/30	50$	60$
b-03	Zelda	Sold	ICantStop	2021/09/24	2021/10/2	70$	100$
b-05	Dress	Sold	ImTooTimid	2021/05/12	2021/05/14	65$	80$

Auction – Category

Category

Product_Type	**Product_Subtype**
Collectibles	BeanieBabies
Outfit	Women

Auction_Category

Bid_ID	**Product_Type**	**Product_Subtype**
b-01	Collectibles	BeanieBabies
b-05	Outfit	Women
b-03	VideoGame	Adventure

Auctions_type

Bid-ID	**Auction_Name**	**Product_Type**	**Product_Subtype**
b-01	Table	Home	Dining
b-03	Blue Ray	Music	CD
b-05	Dress	Outfit	Women

Buyer – Direct_Sell_Product

Direcr_Sell_Product

Product_ID	**Product_Name**	**Product_Amount**	**Product_Price**	**Description**	**Cost_Price**	**Shipping_Type**
s-01	iPhone8	1	280$	Nice condition	500$	Free
s-02	War and Peace	2	50$	Good book	65$	paid

Buyer buys product

Buy_Product

Buyer_ID	**Product_ID**
0004	22-01
0005	22-02

Direct_Sell_Product – Category

Direct_Product_Category

Product_ID	**SellerID**	**Product_Type**	**Product_Subtype**
s-01	0004	Tech	Phone
s-02	0003	Reading	Book

Relational Algebra Expression:

(d) List the item names, item types and their closing prices and winners for all auctions that jjcool has bid on since 12-31-2020.

$$AllBid \leftarrow \Pi_{BidID}(\sigma_{User\_Name=jjcol \wedge Last\_Bid\_Date>'12-31-2020'}(Buyer \bowtie Bid\_History))$$

$$Info \leftarrow \Pi_{Auction.Bid\_ID,Bid.Bid\_Type,Closing\_Price,Winner\_Name}(AllBid \bowtie Bid \bowtie Auction)$$

(f) List the usernames of all bidders who have bid on every auction by IlikePonies.

$$PonyAuctions \leftarrow \Pi_{Bid\_ID}(\sigma_{User\_Name='IlikePonies'}(Bid \bowtie Seller))$$

$$AllBiddersID \leftarrow \Pi_{Bid\_ID,Buyer\_ID}(Bid\_History) \div PonyAuctions$$

$$AllBiddersName \leftarrow \Pi_{User\_Name}(AllBiddersID \bowtie Buyer)$$

(i) List all the usernames and cities/countries of origin of the users who have bid on all the auctions that MileyC has bid on previously (i.e. she had a lower bid on the same auction).

$$BH \leftarrow Bid\_History \bowtie Buyer$$

$$BH2 \leftarrow BH$$

$$AllMileys \leftarrow \Pi_{Bid\_ID,Last\_Bid\_Date}(\sigma_{User\_Name='MileyC'}(BH))$$

$$AllFollowersID \leftarrow \Pi_{Bid\_ID,Buyer\_ID}(BH) \div AllMileys$$

$$AllFollowersInfo \leftarrow \Pi_{User\_Name,City,Country}(AllFollowersID \bowtie Buyer \bowtie Location)$$

(k) List all completed auctions bid on by ICantStop that had a closing price that was either less than half the highest (hidden) proxy bid price he was willing to spend on the auction, or more than twice his highest proxy bid price for the auction.

$$BuyerProxy \leftarrow \sigma_{User\_Name='ICantStop'}(Buyer \bowtie Proxy\_Bidding)$$

$$CompleteBids \leftarrow \sigma_{User\_Name='ICantStop'}(Buyer \bowtie_{Winner\_Name='ICantStop' \wedge Status='Sold'} Auction)$$

$$BidIDs \leftarrow \Pi_{CompleteBids.Bid\_ID}$$

$$(\sigma_{Closing\_Price<0.5*Max\_Price \wedge Closing\_Price>2*Max\_Price \wedge R2.Bid\_ID=R1.Bid\_ID}(\rho_{R1}(CompleteBids) \times \rho_{R2}(Buyerproxy)))$$

$$BidInfo \leftarrow BidIDs \bowtie Bid \bowtie Auction$$

(r) List the userid's of all users who have never bid on auctions of type Collectibles subtype BeanieBabies or any subtype of VideoGame.

$$TypeAuctionsID \leftarrow \Pi_{Bid\_ID}(\sigma_{Product\_Type='Collectibles' \wedge Product\_Subtype='BeanieBabies' \vee Product\_Type='VideoGame'}(Auction\_Category))$$

$$IDBidOn \leftarrow \Pi_{Buyer\_ID}(TypeAuctionsID \bowtie Bid\_Hitory)$$

$$IDNeverBidOn \leftarrow \Pi_{Buyer\_ID}(Buyer) - IDBidOn$$

Part 2 Relational Algebra

2.1

List the name and birth city+state of all artists who have worked on a TVShow episode that was filmed in the city and state of their birth.

Algebra:

$$ArtistLocation \leftarrow \Pi_{ArtistID,ArtistName,BirthLocationID,City,State}(ARTIST \bowtie_{BirthplaceID=LocationID} LOCATION)$$

$$ArtistFilmedPlace \leftarrow \Pi_{ArtistID,LocationID}(WORKED\_ON \bowtie FILMED\_IN)$$

$$WorkonPlace \leftarrow \Pi_{ArtistName,City,State}(ArtistLocation \bowtie_{BirthLocationID=LocationID} ArtistFlimedPlace)$$

Tuple:

$$\{t | \exists u \in Artist(t[ArtsitName] = u[ArtistName] \wedge \exists v \in WORKED\_ON(u[ArtistID] = v[ArtistID]$$

$$\wedge \exists s \in FILMED\_IN(s[EpisodeID] = v[EpisodeID]$$

$$\wedge \exists w \in LOCATION(w[LocationID] = u[BirthLocationID] \vee t[City] = w[City] \wedge t[State] = w[State])))\}$$

2.2

List the name and AID of all artists who have never worked on a TVShow episode that was filmed in the state of their birth and have never worked on an episode that was broadcast in the state of their birth.

Algebra:

$ArtistLocation \leftarrow \Pi_{ArtistID,ArtistName,BirthLocationID}(ARTIST \bowtie_{BirthplaceID=LocationID} LOCATION)$

$EpisodeFilmedPlace \leftarrow \Pi_{ArtistID,LocationID}(WORKED\_ON \bowtie FILMED\_IN)$

$EpisodeBroadcastPlace \leftarrow \Pi_{ArtistID,LocationID}(WORKED\_ON \bowtie BROADCAST\_IN)$

$WorkonPlace \leftarrow \Pi_{ArtistName,ArtistID}(ArtistLocation \bowtie_{BirthLocation=LocationID} EpisodeFilmedPlace \bowtie_{BirthLocation=LocationID} EpisodeBroadcastPlace)$

$NotWorkPlace \leftarrow \Pi_{ArtistID,ArtisName}(ARTIST) - WorkonPlace$

Tuple:

$\{t|\exists u \in ARTIST(t[ArtistName] = u[ArtistName] \wedge t[ArtsitID] = u[ArtistID] \wedge \neg\exists v \in WORKED\_IN(u[ArtistID] = v[ArtsitID]$

$\wedge\exists s \in FILMED\_IN(s[EpisodeID] = v[EpisodeID] \wedge \exists w \in BROADCAST\_IN(w[EpisodeID] = v[EpisodeID]$

$\wedge\exists y \in LOCATION(y[LocationID] = u[BirthLocationID])))))\}$

2.3

List the name of all artists who have worked on every TVShow produced by Sundance studios.

Algebra:

$STVShows \leftarrow \Pi_{ShowID}(\sigma_{StudioName='Sundance'}(TV\_SHOW))$

$ArtistsStudios \leftarrow \Pi_{ArtistName,ShowID}(ARTIST \bowtie WORKDED\_ON \bowtie EPISODE)$

$SundanceArtists \leftarrow ArtistsStudios \div STVShows$

Tuple:

$\{t|\exists u \in ARTIST(t[ArtistName] = u[ArtistName] \wedge \forall v \in TV\_SHOW(v[StudioName] =' Sundance'$

$\rightarrow \exists w \in WORKED\_ON(w[ArtistID] = u[ArtistID] \wedge \exists s \in EPISODE(s[EpisodeID] = w[EpisodeID] \wedge s[ShowID] = v[ShowID])))))\}$

2.4

List the name of all TV shows that have been broadcast in every city in California (doesn't have to be the same episode in all cities).

$CA\_Cities \leftarrow \Pi_{LocationID}(\sigma_{State='California'}(LOCATION))$

$TV\_Locations \leftarrow \Pi_{ShowTitle,LocationID}(TV\_SHOW \bowtie EPISODE \bowtie BROADCAST\_IN)$

$CA\_TVs \leftarrow TV\_Locations \div CA\_Cities$

2.5

List the names and AID's of all artists who have been the lead actor in an episode that they have also directed.

$TwoJob \leftarrow \Pi_{R1.ArtistID}(\sigma_{R1.Job='LeadActor'\wedge R2.Job='Director'}(\rho_{R1}(WORKED\_ON) \times \rho_{R2}(WORKED\_ON)))$

$TwoRoleArtists \leftarrow \Pi_{ArtistName,ArtistID}(Artist \bowtie_{TwoJob.ArtistID=ArtistID} TwoJob)$

2.6

List the names and AID's of all artists who have worked on an episode with someone who worked on an episode with 'Jon Favreau' as director.

Algebra:

$ArtistWork \leftarrow WORKED\_ON \bowtie ARTIST$

$WorkWithJon \leftarrow \Pi_{R2.ArtistID,R2.ArtistName}$

$(\sigma_{R1.ArtistName='JonFavreau'\wedge R1.EpisodeID=R2.EpisodeID\wedge R2.Job='Director'}(\rho_{R1}(ArtistWork) \times \rho_{R2}ArtistWork))$

$AllEpisodes \leftarrow \Pi_{EpisodeID}(WorkWithJon \bowtie_{WorkWithJon.ArtistID=ArtistID} WORKED\_ON)$

$WorkWith \leftarrow \Pi_{ArtistName,ArtistID}(ArtistWork \bowtie AllEpisodes)$

Tuple:

$\{t|\exists u \in ARTIST(t[ArtistName] = u[ArtistName] \wedge t[ArtistID] = u[ArtistID]$

$\wedge \exists v \in WORKED\_ON(v[ArtistID] = u[ArtistID] \wedge \exists s \in ARTIST($

$\exists w \in WORKED\_ON(w[ArtistID] = s[ArtistID] \wedge u[ArtsitName] =' JonFavreau' \wedge w[Job] =' Director' \wedge v[EpisodeID] = w[EpisodeID]$

$\wedge \exists y \in WORKED\_ON(y[ArtistID] = s[ArtistID] \wedge y[EpisodeID] = v[EpisodeID])))))\}$

2.8

List the name of all TV shows that did not earn over a 35.0 rating on every broadcast of the show in Baltimore in 2020.

Algebra:

$ShowsWithRating \leftarrow BROADCAST\_IN \times EPISODE \times TV\_SHOW \times LOCATION$

$less35Shows \leftarrow \Pi_{ShowTitle}(\sigma_{Rating<35.0}(ShowsWithRating))$

$2020BaltimoreShows \leftarrow \Pi_{ShowTitle}(\sigma_{City='Baltimore' \wedge Date<01-01-2021 \wedge Data>12-31-2019}(less35Shows))$

Tuple:

$\{t|\exists u \in TV\_SHOW(T[ShowTitle] = u[ShowTitle]$

$\wedge \forall v \in BROADCAST\_IN(v[Rating] < 35.0 \wedge v[Date] <' 01-01-2021' \wedge v[Date] >' 12-31-2019' \wedge v[EpisodeID] = w[EpisodeID]$

$\wedge \forall s \in LOCATION(s[LocationID] = v[LocationID] \wedge s[City] =' Baltimore' \wedge \rightarrow \exists w \in EPISODE(w[ShowID] = u[ShowID])))\}$

2.9

List the names of all studios that have never produced an episode of a TVshow that has earned more than a 20.0 rating and has never produced an episode that Jennifer Lopez has worked on.

Algebra:

$JLWorkEpisodes \leftarrow \Pi_{EpisodeID}(\sigma_{ArtistName='JenniferLopez'}(ARTIST \bowtie WORKED\_ON))$

$MoreThan20Episodes \leftarrow \sigma_{Rating>20.0}(BROADCAST\_IN \bowtie EPISODE)$

$JLWorkStudio \leftarrow \Pi_{StuidoName}(JLWorkEpisodes \bowtie EPISODE \bowtie TV\_SHOW)$

$MoreThan20Studios \leftarrow \Pi_{StudioName}(MoreThan20Episodes \bowtie TV\_SHOW)$

$NeverStudios \leftarrow \Pi_{StudioName}(TV\_SHOW) - JLWorkStudio - MoreThan20Studios$

tuple:

$\{t|\exists u \in TV\_SHOW(t[StudioName] = u[StudioName] \wedge \neg \exists v \in EPISODE(v[showID] = u[ShowID]$

$\wedge \exists s \in BORADCAST\_IN(s[EpisodeID] = v[EpisodeID] \wedge s[Rating] > 20.0) \wedge \exists w \in ARTIST(w[ArtistName] =' Jenniferlopez'$

$\wedge \exists y \in WORKEED\_ON(y[ArtistID] = w[ArtistID] \wedge v[[EpisodeID] = y[EpisodeID]])))\}$

2.10

List the name of the producing studio (and the episode name) of episodes that have never been broadcast in any location where they were filmed.

Algebra:

$LocationFilmedEpisode \leftarrow \Pi_{EpisodeID}(\sigma_{FILMED\_IN.LocationID=BORADCAST\_IN.LocationID}$

$(FILMED\_IN \bowtie_{FILMED\_IN.EpisodeID=BORADCAST\_IN.EpisodeID} BROADCAST\_IN))$

$StudioAndEpisode \leftarrow \Pi_{EpisodeID,EpisodeTitle,StudioName}(EPISODE \bowtie TV\_SHOW)$

$LocationFilmedStudio \leftarrow StudioAndEpisode \bowtie LocationFilmedEpisode$

$NeverLocationFilmedStudio \leftarrow \Pi_{EpisodeTitle,StudioName}(StudioAndEpisode - LocationFilmedStudio)$

Tuple:

$\{t|\exists u \in TV\_SHOW(t[StudioName] = u[StudioName] \wedge \exists v \in EPISODE(t[EpisodeTitle] = v[EpisodeTitle] \wedge v[ShowID] = u[ShowID]$

$\wedge \neg \exists s \in BORADCAST\_IN(s[EpisodeID] = v[EpisodeID] \wedge \exists w \in FILMED\_IN(w[EpisodeID] = s[EpisodeID]$

$\wedge s[LocationID] = w[LocationID])))\}$

2.11

List all pairs of artists who have worked together on all episodes produced by HBO after 01-01-2021.

$HBOEpisodesAfter2021 \leftarrow \Pi_{EpisodeID}(\sigma_{StudioName='HBO' \wedge CompletionDate>'01-01-2021'}(TV\_SHOW \bowtie EPISODE))$

$WORKED\_ON2 \leftarrow WORKED\_ON$

$ArtistPair \leftarrow \Pi_{WORKED\_ON.ArtistID,WORKED\_ON2.ArtistsID,WORKED\_ON.EpisodeID}$

$(\sigma_{WORKED\_ON.EpisodeID=WORKED\_ON2.EpisodeID}(WORKED\_ON \times WORKED\_ON2))$

$HBOPairs \leftarrow ArtistPair \div HBOEpisodesAfter2021$

2.12

List the TV show name, episode title and episode ID of all episodes in which Jon Favreau was both the director and producer in the same episode

Algebra:

$JonWork \leftarrow \sigma_{ArtistName='JonFavreau'}(Artist \bowtie WORKED\_ON)$

$TwoRoles \leftarrow \Pi_{R1.EpisodeID}(\sigma_{R1.Job='Director' \wedge R2.Job='Producer' \wedge R1.EpisodeID=R2.EpisodeID}(\rho_{R1}(JonWork) \times \rho_{R2}(JonWork)))$

$TVInfo \leftarrow \Pi_{ShowTitle,EpisodeTitle,EpisodeID}(TwoRoles \bowtie EPISODE \bowtie TV\_SHOW)$

Tuple:

$\{t | \exists u \in TV\_SHOW(u[ShowName] = t[ShowName] \wedge \exists v \in EPISODE(v[ShowID] = u[ShowID] \wedge v[EpisodeTitle] = t[EpisodeTitle]$

$\wedge v[EpisodeID] = t[EpisodeID] \wedge \exists w \in ARTIST(w[ArtistName] =' JonFavreau'$

$\wedge \exists s \in WORKED\_ON(s[ArtistID] = w[ArtistID] \wedge \exists y \in WORKED\_ON(y[ArtistID] = w[ArtistID] \wedge y[EpisodeID] = v[EpisodeID]$

$\wedge y[Job] =' Dorector' \wedge w[Job] =' Producer'))))\}$

2.15

List the show name, episode ID and episode title of the highest rated broadcast ever by any TV station in Baltimore. (You cannot use an aggregation operator such as max for this query. Simple relational algebra operators are sufficient).

$NotHignestRate \leftarrow$

$\Pi_{R2.EpisodeID}(\rho_{R1}(BROADCAST\_ON) \bowtie_{R1.Rating>R2.Rating} (\rho_{R2}(BROADCAST\_ON)))$

$HighestRateID \leftarrow \Pi_{EpisodeID}(EPISODE) - NotHighestRate$

$HighestRateInfo \leftarrow$

$\Pi_{ShowTitle,EpisodeID,EpisodeTitle}(HighestRateID \bowtie EPISODE \bowtie TV\_SHOW)$

2.16

List the most expensive “Labor” category (e.g. Stunt Staff) for any episode of Game of Thrones, along with the title of the episode and total amount spent on that category for that episode.

$GTEpisodes \leftarrow$

$\Pi_{EpisodeID,EpisodeTitle}(\sigma_{ShowTitle='GameofThrones'}(TV\_SHOW \bowtie EPISODE))$

$EpisodeSpent \leftarrow GTEpisodes \bowtie SPENT\_ON$

$LaborExpense \leftarrow \sigma_{Broadcategory='Labor'}(ENPENDITURE \bowtie EpisodeSpent)$

$MostExpensive \leftarrow$

$\rho_{r1(maxAmount)}(\mathcal{G}_{max(Amount)}(LaborExpense))$

$ExpenseInfo \leftarrow$

$\Pi_{BroadCategory,EpisodeTitle,Amount}(MostExpensive \bowtie_{MaxAmount=Amount} LaborExpense)$

2.17

List the title and date of all episodes that spent more on Musicians than Actors.

$EpisodeSpent \leftarrow EPISODE \bowtie SPENT\_ON \bowtie EXPENDITURE$

$MusiciansSpenr \leftarrow \sigma_{ExpenseType='Musician'}(EpisodeSpent)$

$ActorsSpent \leftarrow \sigma_{ExpenseType='Actor'}(EpisodeSpent)$

$MoreMusiciansID \leftarrow$

$\Pi_{R1.EpisodeID}(\sigma_{R1.EpisodeID=R2.EpisodeID}(\rho_{R1}MusiciansSpent \bowtie_{R1.Amount>R2.Amount} (\rho_{R2}(ActorsSpent))))$

$EpisodeInfo \leftarrow \Pi_{EpisodeTitle,CompletionDate}(EPISODE \bowtie MoreMusiciansID)$

2.18

List the title and date of all episodes that spent more on Musicians than Actors, along with the title of the TV show containing that episode and the name of the producer of that TV show.

$EpisodeSpent \leftarrow EPISODE \bowtie SPENT\_ON \bowtie EXPENDITURE$

$MusiciansSpenr \leftarrow \sigma_{ExpenseType='Musician'}(EpisodeSpent)$

$ActorsSpent \leftarrow \sigma_{ExpenseType='Actor'}(EpisodeSpent)$

$MoreMusiciansID \leftarrow$

$\Pi_{R1.EpisodeID}(\sigma_{R1.EpisodeID=R2.EpisodeID}(\rho_{R1}MusiciansSpent \bowtie_{R1.Amount>R2.Amount} (\rho_{R2}(ActorsSpent))))$

$ShowProducer \leftarrow \Pi_{ArtistName,EpisodeID}(\sigma_{Job='Producer'}(ARTIST \bowtie WORKED\_ON))$

$ShowWithProducer \leftarrow$

$\Pi_{EpisodeID,CompletionDate,ShowTitle,ArtistName}(ShowProducer \bowtie EPISODE \bowtie TV\_SHOW)$

$MoreMusiciansDetails \leftarrow ShowWithProducer \bowtie MoreMusiciansID$

2.19

List the total amount spent on all types of “Labr” (i.e. a single sum) for each episode of Game of Thrones, along with the title and episode ID of that episode.

$GTEpisodes \leftarrow$

$\Pi_{EpisodeID,EpisodeTitle}(\sigma_{ShowTitle='GameofThrones'}(TV\_SHOW \bowtie EPISODE))$

$EpisodeSpent \leftarrow GTEpisodes \bowtie SPENT\_ON$

$LaborExpense \leftarrow \sigma_{Broadcategory='Labor'}(ENPENDITURE \bowtie EpisodeSpent)$

$TotalSpent \leftarrow$

$\rho_{r1(EpisodeTitle,EpisodeID,TotalAmount)}({}_{EpisodeTitle,EpisodeID}\mathcal{G}_{sum(Amount)}(LaborExpense))$

2.20

List the name of the review sites that have rated every episode of Game of Thrones above 9.0 (¿= 9.0)

$GTEpisodes \leftarrow$

$\Pi_{EpisodeID}(\sigma_{ShowTitle='GameofThrones'}(TV\_SHOW \bowtie EPISODE))$

$AboveSiteScore \leftarrow \sigma_{Score>9.0}(REVIEW\_SITE \bowtie EPISODE\_REVIEW)$

$AboveGTEpisodesSite \leftarrow \Pi_{SiteName,EpisodeID}(AboveSiteScore) \div GTEpisodes$

2.21

List the show title and studio name of the shows which have received a review score of greater than 8.0 for every episode of that show.

$Greater8Episode \leftarrow \sigma_{Score>8.0}(EPISODE\_REVIEW)$

$AllEpisodes \leftarrow \Pi_{EpisodeID,ShowID,StudioName}(EPISODE \bowtie TV\_SHOW)$

$Less8Episode \leftarrow \Pi_{ShowID}(AllEpisodes - \Pi_{EpisodeID}(Greaters8Episodes))$

$Greater8ShowID \leftarrow \Pi_{ShowID}(TV\_SHOW) - Less8Episode$

$Greater8ShowInfo \leftarrow \Pi_{ShowTitle,StudioName}(Greater8ShowID \bowtie TV\_SHOW)$

2.22

List the show title and studio name of the shows which have received a review score of greater than 8.0 for no episode of that show.

$$Less8Episode \leftarrow \sigma_{Score<8.0}(EPISODE\_REVIEW)$$

$$AllEpisodes \leftarrow \Pi_{EpisodeID,ShowID,StudioName}(EPISODE \bowtie TV\_SHOW)$$

$$Greater8Episode \leftarrow \Pi_{ShowID}(AllEpisodes - \Pi_{EpisodeID}(Less8Episodes))$$

$$Less8ShowID \leftarrow \Pi_{ShowID}(TV\_SHOW) - Greater8Episode$$

$$Less8ShowInfo \leftarrow \Pi_{ShowTitle,StudioName}(Less8ShowID \bowtie TV\_SHOW)$$

2.23

List the show title, episode title, broadcast date and broadcast city of all episodes that were broadcast in the same city where they were filmed.

Algebra:

$$SameEpisode \leftarrow$$

$$\Pi_{FILMED\_IN.EpisodeID,FILMED\_IN.LocationID,BROADCAST\_IN.Date}$$

$$(\sigma_{BROADCAST\_IN.Location=FILMED\_IN.Location \wedge BROADCAST\_IN.EpisodeID=FILMED\_IN.EpisodeID}(BORADCAST\_IN \times FILMED\_IN))$$

$$ShowDetails \leftarrow$$

$$\Pi_{EpisodeID,ShowTitle,BROADCAST\_IN.Date,City}(SameEpisode \bowtie EPISODE \bowtie TV\_SHOW \bowtie LOCATION)$$

Tuple:

$$\{t|\exists u \in TV\_SHOW(t[ShoeTitle] = u[ShowTitle] \wedge \exists v \in EPISODE(t[EpisodeTitle] = v[EpisodeTitle]$$

$$\wedge u[ShowID] = v[ShowID] \wedge \exists w \in BROADCAST\_IN(t[Date] = w[Date] \wedge v[EpisodeID] = w[EpisodeID] \wedge$$

$$\exists s \in LOCATION(t[City] = s[City] \wedge s[LocationID] = w[LocationID] \wedge \exists y \in FILMED\_IN(y[LocationID] = w[LocationID]$$

$$\wedge y[EpisodeID] = w[EpisodeID] \wedge s[LocationID] = y[LocationID]))))$$

2.24

List the show title, episode title, expense ID and expense type of all single expenses that exceeded $1,000,000 by themselves (no need to combine/add).

$$ExceedExpense \leftarrow$$

$$\Pi_{EpisodeID,ExpenseID,ExpenseType}(\sigma_{Amount>1000000}(SPENT\_ON \bowtie EXPENDITURE))$$

$$ExceedEpisodes \leftarrow$$

$$\Pi_{EpisodeTitle,ShowTitle,ExpenseID,ExpenseType}(ExceedExpense \bowtie EPISODE \bowtie TV\_SHOW)$$

2.25

List the show title and studio of all shows that had no episode filmed in USA.

$$EpisodeInUSA \leftarrow \Pi_{EpisodeID}(\sigma_{Country='USA'}(FILMED\_IN \bowtie LOCATION))$$

$$InUSAShowID \leftarrow \Pi_{ShowID}(EpisodeInUSA \bowtie EPISODE \bowtie TV\_SHOW)$$

$$NotInUSAShowID \leftarrow \Pi_{ShowID}(TV\_SHOW) - InUSAShowID$$

$$NotInUSAShow \leftarrow \Pi_{ShowTitle,StudioName}(TV\_SHOW \bowtie NotInUSAShowID)$$

2.26

List the name, sex and birth location of all Directors who have worked on the same episode as worked on by Peter Dinklage.

$$ArtistWork \leftarrow ARTIST \bowtie WORKED\_ON$$

$$WorkWith \leftarrow \Pi_{R1.ArtistName,R1.Sex,R1.BirthLocation}$$

$$(\sigma_{R1.EpisodeID=R2.EpisodeID \wedge R2.ArtistName='PeterDrnklage' \wedge R1.Job='Director'}(\rho_{R1}(ArtistWork) \times \rho_{R2}(ArtistWork)))$$

$$WorkWithInfo \leftarrow WorkWith \bowtie_{WorkWith.BirthLocationID=LocationID} LOCATION$$

2.27

List the show title, episode title and episode director of the episode of The Mandalorian which received the highest episode review score on IMDb.

$TMEpisodes \leftarrow \Pi_{ShowTitle,EpisodeID,EpisodeTitle}(\sigma_{ShowTitle='TheMandalorian'}(TV\_SHOW \bowtie EPISODE))$

$IMDbScore \leftarrow$

$\sigma_{SiteName='IMDb'}(TMEpisodes \bowtie EPISODE\_REVIEW \bowtie REVIEW\_SITE)$

$HighestScore \leftarrow$

$\rho_{R1(HighestScore)}(\mathcal{G}_{max(Score)}(IMDbScore))$

$HighestEpisode \leftarrow$

$\Pi_{ShowTitle,EpisodeID,EpisodeTitle}(IMDbScore \bowtie HighestScore)$

$EpisodeInfo \leftarrow$

$\Pi_{ShowTitle,EpisodeTitle,ArtistName}(\sigma_{Job='Director'}(HighestEpisode \bowtie WORKED\_ON \bowtie ARTIST))$

2.28

List the name, sex and birth location of all artists who have either worked on the same episode as worked on by Peter Dinklage, or has worked on the same episode with someone who has worked on the same episode as Peter Dinklage

$ArtistWork \leftarrow ARTIST \bowtie WORKED\_ON$

$WorkWithPD \leftarrow \Pi_{R1.ArtistID}$

$(\sigma_{R1.EpisodeID=R2.EpisodeID \wedge R2.ArtistName='PeterDrnklage'}(\rho_{R1}(ArtistWork) \times \rho_{R2}(ArtistWork)))$

$NewArtistWork \leftarrow ArtistWork \bowtie WorkWithPD$

$WorkAgainWith \leftarrow \Pi_{R1.ArtistID}$

$(\sigma_{R1.EpisodeID=R2.EpisodeID}(\rho_{R1}(ArtistWork) \times \rho_{R2}(NewArtisttWork)))$

$EitherWorkWith \leftarrow WorkWithPD + WorjAgainWith$

$WorkInfo \leftarrow$

$\Pi_{ArtistName,Sex,City,State,Country}(ARTIST \bowtie_{BirthLocationID=LocationID} LOCATION \bowtie EitherWorkWith)$

2.29

List the name of all studios which have a show with at least one episode filmed in the USA.

$EpisodeInUSA \leftarrow \Pi_{EpisodeID}(\sigma_{Country='USA'}(FILMED\_IN \bowtie LOCATION))$

$InUSAShowID \leftarrow \Pi_{ShowID}(EpisodeInUSA \bowtie EPISODE \bowtie TV\_SHOW)$

$InUSAShow \leftarrow \Pi_{StudioName}(TV\_SHOW \bowtie InUSAShow)$

2.30

List the name of all studios which have a show with no episodes filmed in the USA or Canada

Algebra:

$EpisodeInUC \leftarrow \Pi_{EpisodeID}(\sigma_{Country='USA' \vee Country='Canada'}(FILMED\_IN \bowtie LOCATION))$

$InUCShowID \leftarrow \Pi_{ShowID}(EpisodeInUC \bowtie EPISODE \bowtie TV\_SHOW)$

$NotInUCShowID \leftarrow \Pi_{ShowID}(TV\_SHOW) - InUCShowID$

$NotInUCShow \leftarrow \Pi_{StudioName}(TV\_SHOW \bowtie NotInUCShowID)$

Tuple:

$\{t | \exists u \in TV\_SHOW(t[StudioName] = u[StudioName] \wedge \exists v \in TV\_SHOW(v[ShowID] = u[ShowID]$

$\wedge \forall w \in EPISODE(w[ShowID] = v[ShowID] \rightarrow \neg \exists s \in FILMED\_IN(s[EpisodeID] = w[EpisodeID]$

$\exists y \in LOCATION(y[LocationID] = s[LocationID] \wedge (y[Country] =' USA' \vee y[Country] =' Canada'))))))$