

Walmart Sales Database



GROUP MEMBERS:

Sneha Giranje (002785370)

Arundhati Pathrikar (002780632)

Sahil Gothoskar (002775631)

GIT: <https://github.com/SahilGothoskar/Sales/tree/main/Final>

Project Description:

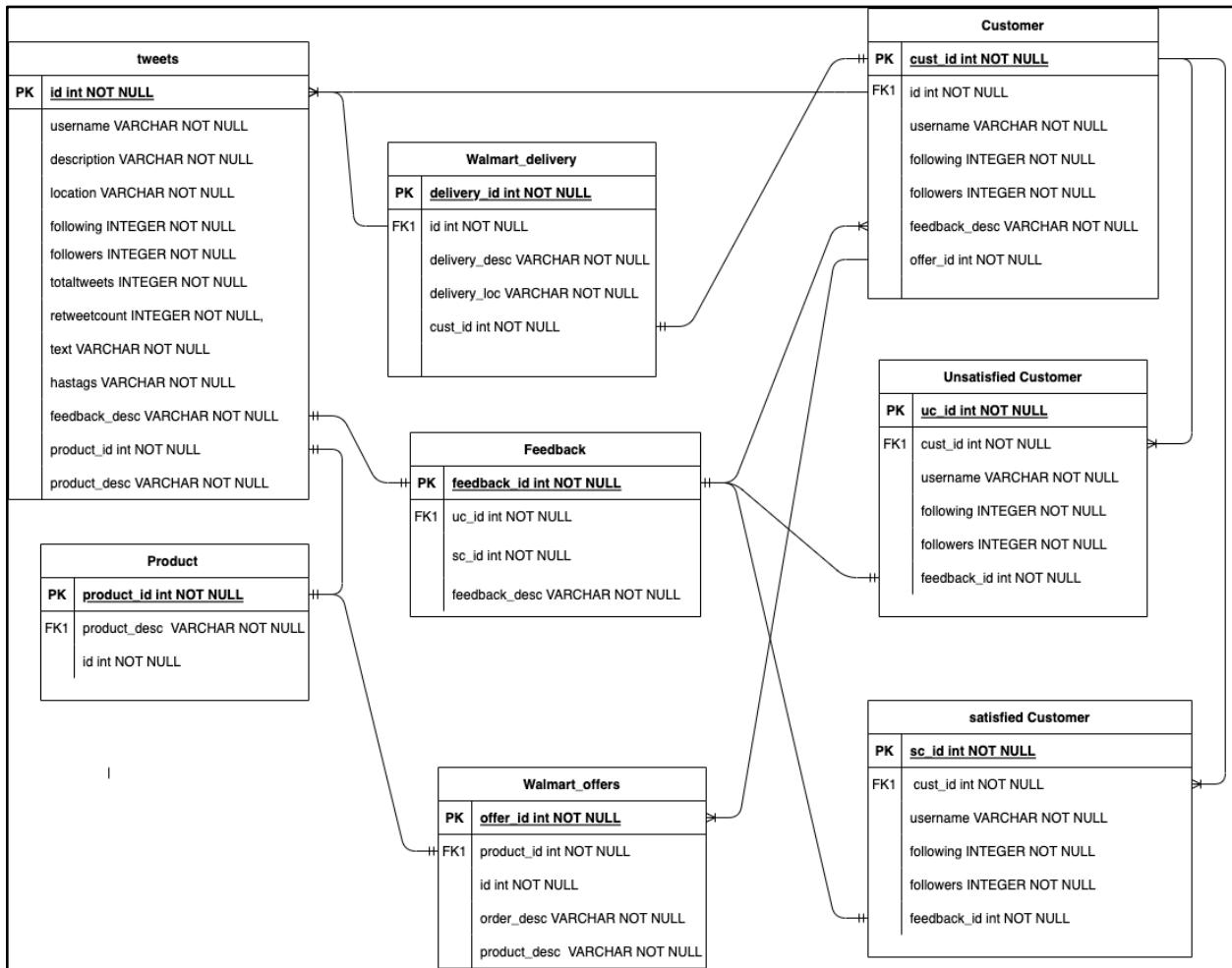
WalMart Stores, Inc., is one of the most well-known and valuable brands in the world. Walmart is one of the leading stores in the United States with about 4,742 stores all over US. There is sales data available for the 45 stores and we would like to predict sales of various stores and how they differ from each other. The dataset should be used to determine which location has the most sales and the largest market capitalization. What do they do differently that helps them increase their sales? Throughout the year, Walmart holds several promotional markdown events. These sales occur before major holidays, the four largest of which are the Super Bowl, Labor Day, Thanksgiving, and Christmas. Weeks with these holidays are weighted five times more heavily in the evaluation than non-holiday weeks. Modeling the effects of markdowns on these holiday weeks in the absence of complete/ideal historical data is part of the challenge presented by this competition. There is historical sales data for 45 Walmart stores in various regions.

Goals:

- Which retailer has the highest sales
- Which retailer's standard deviation is the highest
- Consequently, there are wide variations in sales.
- Likewise, determine the ratio of the mean to the standard deviation.
- Which retailer(s) has/have the best quarterly growth rate in Q3?
- Sales may suffer over certain holidays.
- Find out which holidays have more sales than the average non-holiday season sales for all of the stores combined.
- Give insights and a monthly and semester snapshot of the sales in units.
- A thorough summary of the sales analysis so that other market competitors or new startups can identify the elements that boost sales and the difficulties that depress sales.

UML:

UML Diagram for tables which were generated using data scraped from twitter.



UML Diagram of the tables which were collected , cleaned and munged using python code.

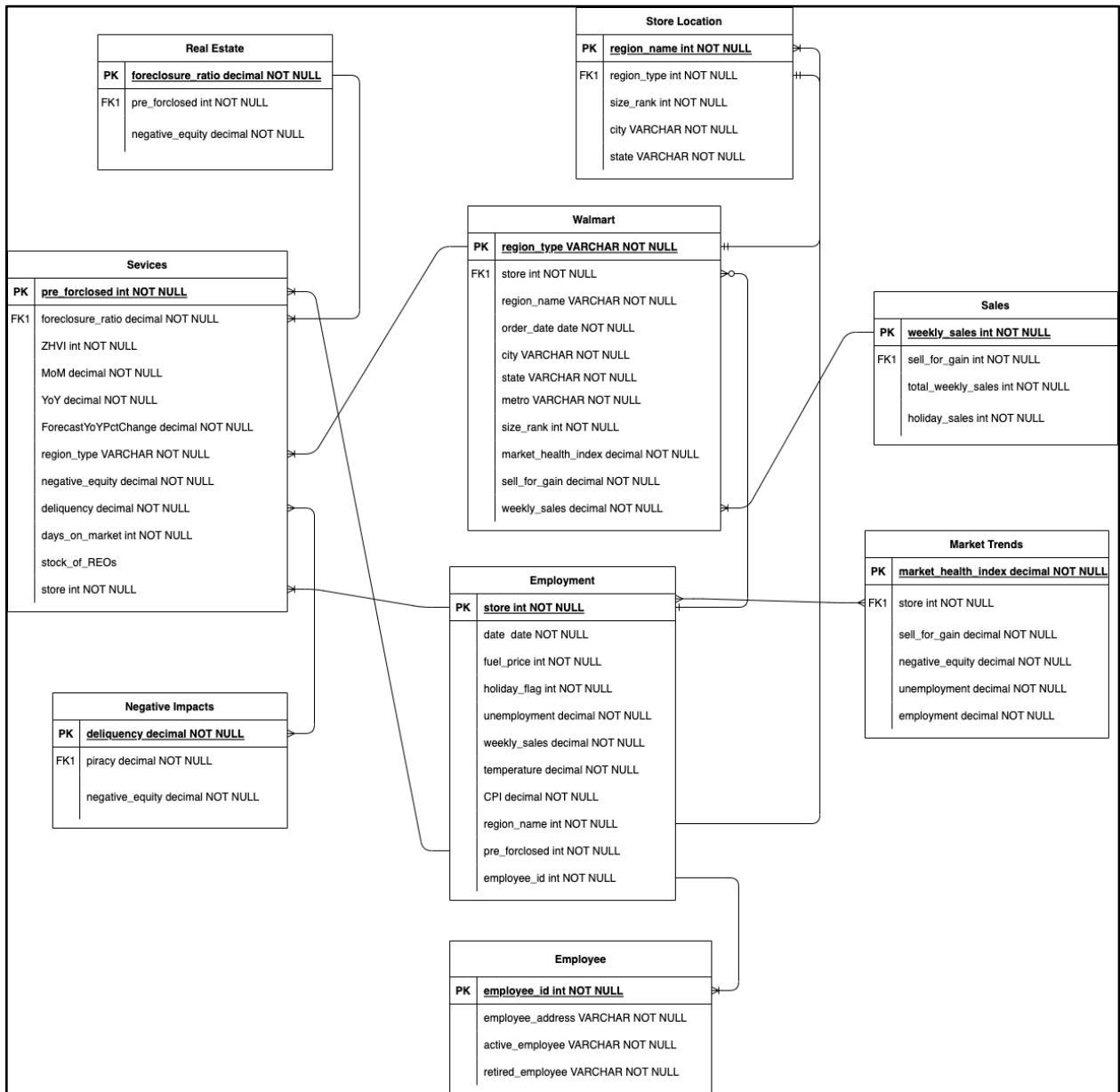


Table Gift Cards:

This table contains information about tweets posted by users with descriptions/hashtags about gift cards.

Table: tweetsgiftcards										
		username	description	location	following	followers	totaltweets	retweetcount	text	hashtags
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1 EmmaAva29	Hi, I am Emma Ava. I have many fre...	New York	52	6	23	0 Get a Free Walmart Gift Card Code ...	[walmartgiftcard', 'giveaways', ...]		
2	2 walmart_newss	Get \$750 Walmart gift cards it is \$10...	united states	87	21	216	1 Walmart \$750 Gift card Giveaway-...	[walmartgiftcard', ...]		
3	3 walmart_newss	Get \$750 Walmart gift cards It Is \$10...	united states	87	21	216	1 Walmart \$750 Gift card Giveaway-...	[walmartgiftcard', ...]		
4	4 LunaElla18	I have an accurately working Free an...	New York	230	15	45	0 How to get a free 500 walmart gift ...	[walmartgiftcard', 'giveaways', ...]		
5	5 ArcGiftcard	Your Chance to get \$750 to your Cas...	NULL	65	22	38	0 Your Chance to win \$100 walmart gl...	[giftcards', 'giveaway', 'free', ...]		
6	6 frankKa63538434	#Free #walmart gift card code ...	NULL	92	46	37	0 Your Chance to win \$100 walmart gi...	[giftcards', 'giveaway', 'free', ...]		
7	7 EmmaMeg93325208	Your Chance to get \$750 to your ...	NULL	154	4	41	0 Your Chance to win \$100 walmart gl...	[giftcards', 'giveaway', 'free', ...]		
8	8 Anthony64303495	Get a \$100 #walmart #giftcard!...	NULL	54	14	51	0 Your Chance to get \$100 walmart gl...	[giftcards', 'giveaway', 'free', ...]		
9	9 AmandaS78118255	ðŸ“Free #cashapp Money generator ...	NULL	166	24	64	0 Your Chance to get \$100 walmart gl...	[giftcards', 'giveaway', 'free', ...]		
10	10 AmandaS78118255	ðŸ“Free #cashapp Money generator ...	NULL	166	24	64	0 See this Instagram photo by ...	[instagram', 'walmart', 'giftcards', ...]		
11	11 Store1Usa	Are you searching for online ...	USA	36	1	2	0 Enter for a \$100 Walmart Gift Card!...	[getawalmartgiftcard', ...]		

Table Tweets:

This table contains information about tweets with descriptions/hashtags about walmart that were posted by users.

		username	description	location	following	followers	totaltweets	veetc...	text	hashtags	createdat
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1 news_cutter	Trusted News Brand in Sri Lanka	NULL		4	2349	7944	0 Walmar...	[walmart']	2021-10-04 17:45:19+00:00	
2	2 amphitritia2	NULL	Neptune		714	229	19399	23 In a ...	[]	2018-06-10 13:13:51+00:00	
3	3 ToddKunzTV	Anchor/Executive Producer ~ ...	Idaho Falls, Idaho		1593	1632	30099	0 #Walm...	[Walmart', ...]	2012-02-03 21:34:08+00:00	
4	4 gordon_riches	NULL	NULL		5113	4855	176420	348 I hear ...	[GunViolence']	2015-07-27 07:33:33+00:00	
5	5 MarcialHyatt6	B.O.I. Born on Island....I am a ...	Galveston Texas		4966	4117	117234	348 I hear ...	[GunViolence']	2019-01-02 11:36:08+00:00	
6	6 sandratxpeach	NULL	NULL		797	890	233237	348 I hear ...	[GunViolence']	2015-04-06 04:58:40+00:00	
7	7 cynical420	#Resistance. #FBR NAFO #FELLAS	NULL		6207	5924	205949	348 I hear ...	[GunViolence']	2014-05-13 18:35:06+00:00	
8	8 grmpamse	Canadian Grampa	NULL		4967	3145	73084	379	[MassShooti...	2020-06-30 18:20:40+00:00	
9	9 amphitritia2	NULL	Neptune		714	229	19399	28 Wake u...	[]	2018-06-10 13:13:51+00:00	
10	10 Shoot_Me_Again	NULL	NULL		880	247	24985	833 #BREA...	[BREAKING', ...]	2009-12-02 18:52:52+00:00	
11	11 HFWrose	dig horror, scifi, music, comics, macs...	Flesh prison Ontari...		113	119	4822	379	[MassShooti...	2009-01-29 05:56:52+00:00	
12	12 Abubaka23635327	Graduate ðŸ—x...	Bahawalpur, Pakistan		1045	158	693	0 Christm...	[Japan', ...]	2020-06-20 07:13:05+00:00	
13	13 mymindtweetsit	Him. ig: ItsMadMike	NULL		336	1119	40270	1 Rip to ...	[Walmart', ...]	2019-01-01 02:13:14+00:00	
14	14 Abubaka23635327	Graduate ðŸ—x...	Bahawalpur, Pakistan		1045	158	693	0 Christm...	[Japan', ...]	2020-06-20 07:13:05+00:00	
15	15 MelissaWillta30	Freelance Marketing Strategist	Portland, OR		206	8	75	0 @Walm...	[Chesapeake...	2011-12-28 19:00:17+00:00	
16	16 JaniceFinneran	Crohn's Disease, Lupus, Fibro, ...	Central Oregon		3124	3194	32242	0 @Walm...	[WalmartSho...	2011-07-14 22:38:59+00:00	
17	17 Abubaka23635327	Graduate ðŸ—x...	Bahawalpur, Pakistan		1045	158	693	0 Christm...	[Japan', ...]	2020-06-20 07:13:05+00:00	
18	18 amphitritia2	NULL	Neptune		714	229	19399	15 https://...]	[]	2018-06-10 13:13:51+00:00	
19	19 Abubaka23635327	Graduate ðŸ—x...	Bahawalpur, Pakistan		1045	158	693	0 Christm...	[Japan', ...]	2020-06-20 07:13:05+00:00	
20	20 Abubaka23635327	Graduate ðŸ—x...	Bahawalpur, Pakistan		1045	158	693	0 Christm...	[Japan', ...]	2020-06-20 07:13:05+00:00	
21	21 Abubaka23635327	Graduate ðŸ—x...	Bahawalpur, Pakistan		1045	158	693	0 Christm...	[Japan', ...]	2020-06-20 07:13:05+00:00	
22	22 Abubaka23635327	Graduate ðŸ—x...	Bahawalpur, Pakistan		1045	158	693	0 Christm...	[Japan', ...]	2020-06-20 07:13:05+00:00	
23	23 YearEightyThree	Black ðŸ—øðŸ‡,			1043	1269	36891	833 #BREA...	[BREAKING', ...]	2013-09-27 17:23:44+00:00	

Table Delivery

This table contains data from user-posted tweets with descriptions/hashtags about Walmart Delivery.

Table: tweetsdelivery										
	id	username	description	location	following	followers	totaltweets	retweetcount	text	hashtags
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	aeroG	Biz/tech/science here. Aviation/spac...	Houston, Texas	2930	2175	77558	3	Four years ago I...	['Walmart', 'Target', 'Alibaba']
2	2	SwardJeff	CEO Merchandising Metrics... ...	Amherst, MA	2235	1491	4963	3	Four years ago I...	['Walmart', 'Target', 'Alibaba']
3	3	therealest007	A girl from the swamp that likes to ta...	Fashion Nation	1604	2326	234159	3	Four years ago I...	['Walmart', 'Target', 'Alibaba']
4	4	michaelzakkour	Founder of '5 NEW DIGITAL' A digital...	New York	2932	2999	16884	3	Four years ago I...	['Walmart', 'Target', 'Alibaba']
5	5	jfeliciano4271	NULL	Palm Harbor, FL	59	13	5	0	Got the new ...	['walmart', 'walmartinhome']
6	6	jpplotsolutions	ðŸ‘©â€œðŸ» Website Development...	worldwide	47	267	725	0	Take Your ...	['Instacart', 'Walmart', ...]
7	7	Cat7t624	â€œmy sons & Doxies. I donâ€™t suffe...	Body: Idiocracy	â€œ:, Asheville	3096	3092	10059	0	occurred: after I spent forever finding
8	8	lavonika	So Below , As Above â€¢ SHOW the ...	@ the DISCOâ€œ, follow me on â€œ+	1167	1120	4240	0	#ShelterLogic ...	['ShelterLogic', 'Walmart']
9	9	Dumidi_bnb	Crypto trader	NULL	20	7	11	1	#Walmartâ€œ #T...	['Walmart', 'Tencent', 'Tigerglobal', ...]
10	10	EmmaOliviajj	SEO & Marketing Expert at Vouchers...	New York	3	1	42	0	Walmart Promo ...	['vouchersportalus', 'couponscode', ...]
11	11	NikkicCo4351208	Samoan/native American, 1984 from ...	Seneca, SC	1895	406	1418	0	Wow now I ...	['canceled', 'Seneca', 'SouthCarolina']
12	12	nerdharder88	Arkansas - TV & Movies - Comic Book...	Arkansas, USA	576	439	4142	0	@walmart ...	['walmart', 'spark', 'walmartspark', ...]
13	13	nerdharder88	Arkansas - TV & Movies - Comic Book...	Arkansas, USA	576	439	4142	0	@Walmart ...	['walmart', 'spark', 'walmartspark', ...]
14	14	nerdharder88	Arkansas - TV & Movies - Comic Book...	Arkansas, USA	576	439	4142	0	@walmart ...	['walmart', 'spark', 'walmartspark', ...]
15	15	genpooh	Teal stage 3 Ovarian Cancer Survivor ...	Alabama, USA	1105	709	32573	0	My personal ...	['Walmart']
16	16	littlefobat	NULL	NULL	75	20	34	0	Do not ever ma...	['walmart', 'walmartgrocery']
17	17	skpanwala	NULL	Toronto, Ontario	491	67	15	0	#walmartBlackF...	['walmartBlackFriday', 'Walmart', ...]
18	18	swiperheight10	I am proud I have no followers on ...	NULL	10	2	2497	0	@AmazonHelp ...	['Walmart']
19	19	KristenD0819	MS Warrior ðŸši ...	North Carolina, USA	278	130	3244	0	On Friday I plac...	['Walmart', 'walmart']
20	20	ramirezjoseph24	NULL	NULL	138	11	689	0	@Walmart ...	['walmart', 'walmartBlackFriday', ...]
21	21	SolKristaK	Patriot, Sky Watcher, Retired Disable...	SoCal	781	234	2875	0	#Walmart...	['Walmart']

Table Fart

This table contains data from user-posted tweets with customers who are pissed with walmart descriptions/hashtags.

Table: tweetsfart									
	id	username	description		location	following	followers	totaltweets	retweetcount
	Filter	Filter	Filter		Filter	Filter	Filter	Filter	Filter
1	1	Jaquavios_Clark	Wezer	NULL		361	48	2183	0 @Walmart_QnADept @DakeRapper ...
2	2	theTEAfascist	goof...	California sf		851	602	30843	0 @jasperhaTsgf WALFART
3	3	SilversDarrin	NULL	NULL		107	30	160	0 @GigaBeers @brixwe I'm wondering ...
4	4	Painoman1016	I like music a lot, probably gonna pos...	NULL		100	24	959	214 Sena goes to Walfart and goes to the...
5	5	dsprad117	VFL ...	Sylvania, OH		1180	108	1215	0 @jusnhan_ These are thee best coug...
6	6	SammySlipschit	NULL	Redwood country, California		535	32	11987	0 @Willieslille2 @karawisher Oh, wha...
7	7	NHarris956	Father & Grandfather. öýşgiöý' Despi...	Colorado, USA		10143	9828	10801	0 @MadScientistFF Walfart
8	8	Quinlot1	Comic cons, comics, anything ...	NULL		386	266	2053	0 @Kaylan_TX Only at Walfart öýxöÝk...
9	9	Byx_Xeno	SFW & mild NSFW I draw Xenobla...	My room		163	975	3201	0 @amandapena1020 NOOO I PAID FO...
10	10	AkujiSeth	İâ€™m the monster under your bed.	Under ur bed, below ur head		895	363	1779	0 And I broke down in tears. Idk why I ...
11	11	AkujiSeth	İâ€™m the monster under your bed.	Under ur bed, below ur head		895	363	1779	0 @haziliz I work at walfart...

Employment Table:

Employment Table contains different store details like their weekly sales, unemployment percent etc

- Primary key for this table is Employment_Id
- Foreign key for this table is Services_ID

The table is normalized and also cleaned.

The screenshot shows a database interface with a sidebar titled 'SCHEMAS' containing 'shopping_model', 'sys', and 'walmart'. Under 'walmart', there are 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'employment' table is selected under 'Tables'. The main area is a 'Result Grid' showing data for the 'employment' table. The columns are: Store, Date, Weekly_Sales, Holiday_Flag, Temperature, Fuel_Price, CPI, Unemployment, and Employment_ID. The data consists of 32 rows, each representing a different date from February 2010 to August 2010, with various numerical values for each column. The 'employment' table is also visible at the bottom of the grid.

Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	CPI	Unemployment	Employment_ID
1	05-02-2010	1643690.9	0	42.31	2.572	211.0963582	8.106	1
1	28-10-2011	1445249.09	0	66.57	3.372	217.6766791	7.866	2
4	02-07-2010	1881337.21	0	73.66	2.668	126.1392	7.372	3
2	13-07-2012	1830075.13	0	80.17	3.256	221.5616784	6.565	4
1	12-02-2010	1641957.44	1	38.51	2.548	211.2421698	8.106	5
1	19-02-2010	1611968.17	0	39.93	2.514	211.2891429	8.106	6
1	26-02-2010	1409727.59	0	46.63	2.561	211.3196429	8.106	7
1	05-03-2010	1554806.68	0	46.5	2.625	211.3501429	8.106	8
1	12-03-2010	1439541.59	0	57.79	2.667	211.3806429	8.106	9
1	19-03-2010	1472515.79	0	54.58	2.72	211.215635	8.106	10
1	26-03-2010	1404429.92	0	51.45	2.732	211.0180424	8.106	11
1	02-04-2010	1594968.28	0	62.27	2.719	210.8204499	7.808	12
1	09-04-2010	1545418.53	0	65.86	2.77	210.6228574	7.808	13
1	16-04-2010	1466058.28	0	66.32	2.808	210.4887	7.808	14
1	23-04-2010	1391256.12	0	64.84	2.795	210.4391228	7.808	15
1	30-04-2010	1425100.71	0	67.41	2.78	210.3895456	7.808	16
1	07-05-2010	1603955.12	0	72.55	2.835	210.3399684	7.808	17
1	14-05-2010	1494251.5	0	74.78	2.854	210.3374261	7.808	18
1	21-05-2010	1399662.07	0	76.44	2.826	210.6170934	7.808	19
1	28-05-2010	1432069.95	0	80.44	2.759	210.8967606	7.808	20
1	04-06-2010	1615524.71	0	80.69	2.705	211.1764278	7.808	21
1	11-06-2010	1542561.09	0	80.43	2.668	211.4560951	7.808	22
1	18-06-2010	1503284.06	0	84.11	2.637	211.4537719	7.808	23
1	25-06-2010	1422711.6	0	84.34	2.653	211.3386526	7.808	24
1	02-07-2010	1492418.14	0	80.91	2.669	211.2235333	7.787	25
1	09-07-2010	1546074.18	0	80.48	2.642	211.108414	7.787	26
2	20-07-2012	1819666.46	0	83.23	3.311	221.5701123	6.565	27
1	16-07-2010	1448938.92	0	83.15	2.623	211.1003854	7.787	28
1	23-07-2010	1385065.2	0	83.36	2.608	211.2351443	7.787	29
1	30-07-2010	1371986.6	0	81.84	2.64	211.3699032	7.787	30
1	06-08-2010	1605491.78	0	87.16	2.627	211.5046621	7.787	31
1	13-08-2010	1508237.76	0	87	2.692	211.6394211	7.787	32

Services Table:

Services Table contains different store details like their ZHVI ,MoM, YoY

- Primary key for this table is Services_ID
- Foreign key for this table is Employment_Id

The table is normalized and also cleaned.

	ZHVI	MoM	YoY	ForecastYoY	PctChng	NegativeEquity	Delinquency	DaysOnMarket	Services_ID
> shopping_model									
> sys									
< walmart									
Tables									
> employment									
> services									
> walmart									
Views									
Stored Procedures									
Functions									
services 19									
	633400	0.300870942201108	7.1017923571187	0.017052415535207	0.05447241395301	0.031303563006052	66	1	
	185700	1.47540983606557	13.717085194121	0.03751211631664	0.12871534834169	0.032028471577531	64	2	
	324000	0.598978919580189	4.11311053984576	0.005450617283951	0.071275671505588	0.046848436870027	102	3	
	223300	0.359550561797753	7.30418068236425	0.033954321540529	0.228913937602891	0.040972971703156	84	4	
	469900	0.707243891984589	6.8926296633303	0.0354117897425	0.083345494074341	0.038472460571585	59	5	
	140300	0.10791936645086	7.18105423987777	0.032815395580898	0.148186994559375	0.074888506376121	84	6	
	222000	0.13531799729364	5.36307546274524	0.009418918918919	0.068463434087399	0.050126263458889	110	7	
	218600	0.598251265531523	9.88389362769694	0.038069533394328	0.112508516965411	0.020692503126331	64	8	
	541700	0.3705762460626287	6.2156862745088	-0.001140652870593	0.042395981639399	0.009179905724258	46	9	
	227000	1.02358700489542	11.3837095191364	0.05092026431718	0.162197221929523	0.064401786312163	60	10	
	121800	0.082169288693509	1.07883817427386	0.036633825944171	0.09489722001514	0.045184793912029	60	11	
	586800	0.427862399452336	7.92716571638771	0.032987389229721	0.063725471831529	0.015708405336935	51	12	
	323700	0.465549348230912	6.86895278969957	0.028260735248687	0.063546797214152	0.036843957739265	61	13	
	170800	1.12492599171107	9.82857142857143	0.048495316159251	0.063725471831529	0.015708405336935	62	14	
	912300	0.784357048166151	9.32294787297783	0.030925134276006	0.030384420274499	0.011059087612744	40	15	
	177800	0.736543909348442	8.87936313533374	0.047975253093363	0.07307229478761	0.031638173653825	58	16	
	154900	0.388852683992223	10.4062722736992	0.042724338282763	0.162267910008616	0.061307503786058	75	17	
	259100	0.699572483482316	11.8256365990505	0.048097259745272	0.073259031482079	0.040297317718704	49	18	
	12367...	0.471199935006905	10.9546025477993	0.0234313924153	0.034161957770519	0.008251978925318	44	19	
	135100	0.595681310498883	10.6470106470106	0.051628423390081	0.066470401567977	0.016734360365345	50	20	
	327000	0.337526848726603	8.27814569536424	0.032840978593272	0.034161957770519	0.008251978925318	57	21	
	494700	0.712540716612378	5.75032064985036	0.029575000303214	0.064602403068049	0.021748963750399	53	22	
	47700	4.60526315789474	33.9887640449438	0.09	0.361762843797311	0.040241319093584	109	23	
	164400	0.366300366300366	6.40776699029126	0.023248175182482	0.111582703070531	0.060752175430819	73	24	
	131700	0.765110941086458	5.87378400659522	0.042414578587699	0.116894165221145	0.040414193776267	54	25	
	303800	-0.197109067017083	3.22799864084268	0.026349572086899	0.141190817882697	0.035016159179671	60	26	
	79300	-1.85643564356436	2.85343709468223	0.03960979445145	0.149323320834595	0.062136060920286	78	27	
	128800	0.625	10.5579399141631	0.046048136645963	0.141037322611783	0.061298116294996	101	28	
	190900	0.898520084566598	9.52380952380952	0.04429024295443	0.103314848331934	0.03140262653822	53	29	
	775400	0.453426609684464	5.03928474668112	0.010483621356719	0.05268574029939	0.01203049671752	57	30	
	113600	-0.35087719292450	0.8201771477918...	0.025536971830966	0.103314848331934	0.03140262653822	91	31	
	219700	0.457247370827618...	9.0325080451613...	0.02893946290396	0.10508133780295	0.006666823419092	49	32	

Walmart Table:

Walmart table contains different stores across different regions along with their market health index, size rank and sale for gain

- Primary key for this table is Walmart_Id

The table is normalized and also cleaned.

	RegionType	RegionName	City	State	Metro	SizeRank	MarketHealthIndex	SellForGain	Walmart_id
> shopping_model									
> sys									
✓ walmart									
└ Tables									
> employment	City	Chesapeake	Chesapeake	VA	Virginia Beach, VA	86	1.03486037598102	78.52	1
	City	Los Angeles	Los Angeles	CA	Los Angeles-Long Beach-Anaheim, CA	1	7.5552108048914	96.37	2
	City	Hialeah	Hialeah	FL	Miami-Fort Lauderdale, FL	87	8.6722095674393	89.66	3
	City	Chicago	Chicago	IL	Chicago, IL	2	4.10293849242562	81.92	4
	City	Madison	Madison	WI	Madison, WI	88	9.09746304070086	93.23	5
	City	Philadelphia	Philadelphia	PA	Philadelphia, PA	3	3.39569264464318	87.6	6
	City	Fontana	Fontana	CA	Riverside, CA	89	7.56889943420332	91.4	7
	City	Fort Myers	Fort Myers	FL	Fort Myers, FL	90	6.32049644095638	85.71	8
	City	Phoenix	Phoenix	AZ	Phoenix, AZ	4	8.48147472168454	88.21	9
	City	Knoxville	Knoxville	TN	Knoxville, TN	91	7.87734988138521	100	10
	City	Laredo	Laredo	TX	Laredo, TX	92	7.25679868589159	100	11
	City	Las Vegas	Las Vegas	NV	Las Vegas, NV	5	5.55119547362657	81.94	12
	City	San Diego	San Diego	CA	San Diego, CA	6	8.76528563606498	96.79	13
	City	Dallas	Dallas	TX	Dallas-Fort Worth, TX	7	9.88684066435481	96.65	14
	City	North Las V...	North Las...	NV	Las Vegas, NV	93	4.24164993611973	78.61	15
	City	Lawrenceville	Lawrenceville	GA	Atlanta, GA	94	8.07081584230699	100	16
	City	San Jose	San Jose	CA	San Jose, CA	8	9.61489322869137	97.97	17
	City	Jacksonville	Jacksonville	FL	Jacksonville, FL	9	4.53276145281986	80.76	18
	City	San Francis...	San Franci...	CA	San Francisco, CA	10	9.776419054572	98.6	19
	City	Austin	Austin	TX	Austin, TX	11	8.68771673663077	100	20
	City	Fremont	Fremont	CA	San Francisco, CA	95	8.2870961854353	100	21
	City	Detroit	Detroit	MI	Detroit, MI	12	9.08924986311371	100	22
	City	Fayetteville	Fayetteville	NC	Fayetteville, NC	96	4.17868224128491	50	23
	City	Columbus	Columbus	OH	Columbus, OH	13	3.84102938492426	85.8	24
	City	Memphis	Memphis	TN	Memphis, TN	14	1.10147837196568	80.73	25
	City	Springfield	Springfield	MO	Springfield, MO	97	1.65358642087972	91.67	26
	City	Charlotte	Charlotte	NC	Charlotte, NC	15	6.22741376163533	91.03	27
	City	Arlington	Arlington	VA	Washington, VA	98	7.1436393502464	91.67	28
	City	El Paso	El Paso	TX	El Paso, TX	16	0.3896696477459...	0	29
	City	Irving	Irving	TX	Dallas-Fort Worth, TX	99	9.76181784997262	100	30
	City	Boston	Boston	MA	Boston, MA	17	8.58733345501004	97.69	31
	City	Gilbert	Gilbert	AZ	Phoenix, AZ	100	7.2093447709456	90.82	32

Major use cases & output snippets

Local instance 3306 Local instance 3306 (walmart)

Administration Schemas Query 1

SCHEMAS Filter objects

- > shopping_model
- > sys
- walmart
 - Tables
 - Views
 - Stored Procedures
 - Functions

Result Grid Filter Rows: Search: Export:

```

1 • use walmart;
2 • SELECT Store , SUM(Weekly_Sales) as Total_weeklysales FROM employment GROUP BY Store;
3

```

100% 1:3

Result Grid Result Grid Form Editor Field Types Query Stats Execution Plan

Store	Total_weeklysales
2	275382440.9800001
3	57586735.07
4	298990953.38
5	4547568.0
6	22376130.64000002
7	81598275.14
8	129991181.3
9	77111785.99
10	271617713.8999999
11	193962786.79999992
12	144287238.4999999
13	298990953.38
14	288990911.34000003
15	89133683.91999997
16	74252425.40000002
17	12776130.64000007
18	155114734.21
19	206634862.1
20	301397792.40000004
21	161117875.92
22	147075648.57000002
..	..
Result 2	..

Action Output

Time	Action	Response	Duration / Fetch Time
1	18:22:48 use walmart	0 row(s) affected	0.00030 sec
2	18:22:48 SHOW FULL TABLES WHERE table_type = "VIEW"	15 row(s) returned	0.0051 sec / 0.00001...
3	18:38:31 SELECT Store , SUM(Weekly_Sales) as Total_weeklysales FROM employment GROUP BY Stor...	45 row(s) returned	0.0096 sec / 0.00001...

Query Completed

Local instance 3306 Local instance 3306 (walmart)

Administration Schemas Query 1

SCHEMAS Filter objects

- > shopping_model
- > sys
- walmart
 - Tables
 - Views
 - Stored Procedures
 - Functions

Result Grid Filter Rows: Search: Export: Fetch rows:

```

1 • use walmart;
2 • SELECT Store , Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 AND Fuel_Price > 3.5;
3

```

100% 1:3

Result Grid Result Grid Form Editor Field Types Query Stats Execution Plan

Store	Date	Fuel_Price	Holiday_Flag
1	01-04-2011	3.524	0
1	08-04-2011	3.622	0
1	15-04-2011	3.743	0
1	22-04-2011	3.807	0
1	29-04-2011	3.81	0
1	06-05-2011	3.695	0
1	13-05-2011	3.699	0
1	20-05-2011	3.607	0
1	27-05-2011	3.766	0
1	03-06-2011	3.699	0
1	10-06-2011	3.648	0
1	17-06-2011	3.637	0
1	24-06-2011	3.594	0
1	01-07-2011	3.524	0
1	18-07-2011	3.675	0
1	25-07-2011	3.651	0
1	29-07-2011	3.682	0
1	05-08-2011	3.684	0
1	12-08-2011	3.638	0
1	19-08-2011	3.554	0
1	26-08-2011	3.523	0

Action Output

Time	Action	Response	Duration / Fetch Time
1	18:22:48 use walmart	0 row(s) affected	0.00030 sec
2	18:22:48 SHOW FULL TABLES WHERE table_type = "VIEW"	15 row(s) returned	0.0051 sec / 0.00001...
3	18:38:31 SELECT Store , Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 A...	45 row(s) returned	0.0096 sec / 0.00001...
4	18:43:17 SELECT Store , Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 A...	1000 row(s) returned	0.0032 sec / 0.0012 s...

Query Completed

Local instance 3306 Local instance 3306 (walmart)

Administration Schemas Query 1

```

1 • use walmart;
2 • SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales BETWEEN 1542561.09 AND 1606629.58 GROUP BY Store;
3
4

```

Result Grid Filter Rows: Search Export:

MIN(Unemployment)	Store
6.573	1

Result 4 Action Output

Action	Time	Response	Duration / Fetch Time
use walmart	18:22:48	0 row(s) affected	0.00030 sec
SHOW FULL TABLES WHERE table_type = "VIEW"	18:22:48	15 row(s) returned	0.0051 sec / 0.0001...
SELECT Store , SUM(Weekly_Sales) as Total_weeklysales FROM employment GROUP BY Store LIMIT 0, 1000	18:38:31	45 row(s) returned	0.0096 sec / 0.0001...
SELECT Store, Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 AND Fuel_Price > 3.5 LIMIT 0, 1000	18:43:17	1000 row(s) returned	0.0032 sec / 0.0012 s...
SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales BETWEEN 1542561.09 AND 1606629.58...	18:44:32	1 row(s) returned	0.0086 sec / 0.0001...

Query Completed

Local instance 3306 Local instance 3306 (walmart)

Administration Schemas Query 1

```

1 • use walmart;
2 • SELECT AVG(Weekly_Sales), Store from employment GROUP BY Store;
3

```

Result Grid Filter Rows: Search Export:

AVG(Weekly_Sales)	Store
1555261.3875524479	1
1925751.3355244761	2
402704.44104895106	3
2094712.9000993007	4
3186000.0000000000	5
1564729.1860370303	6
570617.3095713287	7
900749.5183916084	8
543980.5523776223	9
1894424.572657342	10
1356383.1244755238	11
1009001.6694405593	12
2002000.0000000000	13
2009919.4009790261	14
623212.4749650348	15
519247.7300099302	16
893581.390419581	17
1084718.421048951	18
1444999.0356643356	19
2107876.8703496507	20
756069.0833566433	21

Result 5 Action Output

Action	Time	Response	Duration / Fetch Time
use walmart	18:22:48	0 row(s) affected	0.00030 sec
SHOW FULL TABLES WHERE table_type = "VIEW"	18:22:48	15 row(s) returned	0.0051 sec / 0.0001...
SELECT Store , SUM(Weekly_Sales) as Total_weeklysales FROM employment GROUP BY Store LIMIT 0, 1000	18:38:31	45 row(s) returned	0.0096 sec / 0.0001...
SELECT Store, Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 AND Fuel_Price > 3.5 LIMIT 0, 1000	18:43:17	1000 row(s) returned	0.0032 sec / 0.0012 s...
SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales BETWEEN 1542561.09 AND 1606629.58...	18:44:32	1 row(s) returned	0.0086 sec / 0.0001...
SELECT AVG(Weekly_Sales), Store from employment GROUP BY Store LIMIT 0, 1000	18:45:13	45 row(s) returned	0.010 sec / 0.00015...

Query Completed

Local instance 3306 Local instance 3306 (walmart)

Administration **Schemas** **Query 1**

SCHEMAS **Filter objects**

- > shopping_model
- > sys
- < **walmart**
 - > Tables
 - > Views
 - > Stored Procedures
 - > Functions

100% | 1:3 |

Result Grid **Filter Rows:** **Search:** **Export:**

Store	SUM(Holiday_Flag)
1	10
2	10
3	10
4	10
5	10
6	10
7	10
8	10
9	10
10	10
11	10
12	10
13	10
14	10
15	10
16	10
17	10
18	10
19	10
20	10
21	10

Result 6

Action Output

Action	Time	Response	Duration / Fetch Time
use walmart	18:22:48	0 row(s) affected	0.00030 sec
SHOW FULL TABLES WHERE table_type = 'VIEW'	18:22:48	15 row(s) returned	0.0051 sec / 0.00001...
SELECT Store , SUM(Weekly_Sales) as Total_WeeklySales FROM employment GROUP BY Store LIMIT 0, 1000	18:38:31	45 row(s) returned	0.0096 sec / 0.00001...
SELECT Store, Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 AND Fuel_Price > 3.5 LIMIT 0, 1000	18:43:17	1000 row(s) returned	0.0032 sec / 0.0012 s...
SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales BETWEEN 1542561.09 AND 1606629.58...	18:44:32	1 row(s) returned	0.0086 sec / 0.00001...
SELECT AVG(Weekly_Sales), Store from employment GROUP BY Store LIMIT 0, 1000	18:45:13	45 row(s) returned	0.010 sec / 0.000015...
SELECT Store, SUM(Holiday_Flag) FROM employment GROUP BY Store HAVING SUM(Holiday_Flag) >9 LIMIT 0, 1000	18:45:51	45 row(s) returned	0.019 sec / 0.000013...

Query Completed

Local instance 3306 Local instance 3306 (walmart)

Administration **Schemas** **Query 1**

SCHEMAS **Filter objects**

- > shopping_model
- > sys
- < **walmart**
 - > Tables
 - > Views
 - > Stored Procedures
 - > Functions

100% | 1:4 |

Result Grid **Filter Rows:** **Search:** **Export:**

RegionName	City	DaysOnMarket
Phoenix	Phoenix	46
Phoenix	Phoenix	141
Phoenix	Phoenix	83

Result 7

Action Output

Action	Time	Response	Duration / Fetch Time
use walmart	18:22:48	0 row(s) affected	0.00030 sec
SHOW FULL TABLES WHERE table_type = 'VIEW'	18:22:48	15 row(s) returned	0.0051 sec / 0.00001...
SELECT Store , SUM(Weekly_Sales) as Total_WeeklySales FROM employment GROUP BY Store LIMIT 0, 1000	18:38:31	45 row(s) returned	0.0096 sec / 0.00001...
SELECT Store, Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 AND Fuel_Price > 3.5 LIMIT 0, 1000	18:43:17	1000 row(s) returned	0.0032 sec / 0.0012 s...
SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales BETWEEN 1542561.09 AND 1606629.58...	18:44:32	1 row(s) returned	0.0086 sec / 0.00001...
SELECT AVG(Weekly_Sales), Store from employment GROUP BY Store LIMIT 0, 1000	18:45:13	45 row(s) returned	0.010 sec / 0.000015...
SELECT Store, SUM(Holiday_Flag) FROM employment GROUP BY Store HAVING SUM(Holiday_Flag) >9 LIMIT 0, 1000	18:45:51	45 row(s) returned	0.019 sec / 0.000013...
SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id where City=...	18:51:50	3 row(s) returned	0.014 sec / 0.000012...

Query Completed

The screenshot shows a database management interface with the following details:

- Top Bar:** Local instance 3306, Local instance 3306 (walmart), Administration, Schemas, Query 1.
- Schemas Panel:** shopping_model, sys, walmart (selected), Tables, Views, Stored Procedures, Functions.
- Query Editor:** A code editor with the following SQL query:

```
1 • use walmart;
2 • SELECT walmart.SizeRank, walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON walmart.Walmart_id = services.services
3 |
```
- Result Grid:** Shows one row of data:

SizeRank	RegionName	City	DaysOnMarket
13098	Morganton	Morganton	106
- Execution Plan:** A detailed table showing the execution steps, time taken, and response for each step. The last step is highlighted in green.
- Bottom Status:** No object selected, Read Only.

The screenshot shows the MySQL Workbench application. At the top, there are two tabs: "Local instance 3308" and "Local instance 3308 (walmart)". The main window has several panes:

- Administration**: Shows connection details.
- Schemas**: A tree view of database structures under "walmart".
- Query Editor**: Contains the following SQL code:

```
1 • use walmart;
2 • SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency FROM walmart INNER JOIN services ON walmart.Walmart_id = services.service_id;
```
- Result Grid**: Displays the results of the query in a tabular format. The columns are RegionName, City, NegativeEquity, and Delinquency. The data includes rows for Des Moines, Atlanta, Toledo, Gainesville, McKinney, West Valley City, Renton, Beaverton, Joliet, Albany, Highlands Ranch, Elkhart, Evanston, Shelby Township, Miami Beach, Herndon, Pawtucket, Johnston City, Casa Grande, Warwick, and Palo Alto.
- Object Info**: Shows no objects selected.
- Session**: Shows no session information.
- Execution Plan**: Located on the right side, showing the execution plan for the query.

Local instance 3306 Local instance 3306 (walmart)

Administration Schemas Query 1

SCHEMAS Filter objects

- shopping_model
- sys
- walmart
 - Tables
 - Views
 - Stored Procedures
 - Functions

100% 13 | Result Grid Filter Rows: Search Export: Fetch rows: 1000 rows

RegionName	City	DaysOnMarket
Chesapeake	Chesapeake	66
Los Angeles	Los Angeles	64
Hartford	Hartford	62
Chicago	Chicago	84
Madison	Madison	59
Philadelphia	Philadelphia	84
Fontana	Fontana	110
Fort Myers	Fort Myers	64
Phoenix	Phoenix	46
Knoxville	Knoxville	60
Laredo	Laredo	68
Las Vegas	Las Vegas	51
San Diego	San Diego	61
Dallas	Dallas	62
North Las V...	North Las...	40
Lawrenceville	Lawrenceville	58
San Jose	San Jose	75
Jacksonville	Jacksonville	49
San Francisco...	San Frans...	44
Austin	Austin	50
Fremont	Fremont	57

No object selected Object Info Session Read Only

Result 10

Action Output

Time	Action	Response	Duration / Fetch Time
18:44:32	SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales BETWEEN 1542561.09 AND 1606629.58...	1 row(s) returned	0.0086 sec / 0.00001...
18:45:13	SELECT AVG(Weekly_Sales), Store from employment GROUP BY Store LIMIT 0, 1000	45 row(s) returned	0.010 sec / 0.000015...
18:45:51	SELECT Store, SUM(Holiday_Flag) FROM employment GROUP BY Store HAVING SUM(Holiday_Flag) >9 LIMIT 0, 1000	45 row(s) returned	0.019 sec / 0.000013...
18:51:50	SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON walmart.Walmart_id = s...	3 row(s) returned	0.014 sec / 0.000012...
18:52:23	SELECT walmart.SizeRank, walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON wal...	1 row(s) returned	0.020 sec / 0.000026...
18:53:12	SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from walmart INNER JOIN services ON w...	1000 row(s) returned	0.0083 sec / 0.0094...
18:53:54	SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart LEFT JOIN services ON walmart.Walmart_id = s...	1000 row(s) returned	0.0038 sec / 0.0017 s...

Query Completed

SQL 1

```
1 SELECT COUNT(*) FROM tweets WHERE hashtags LIKE "%walmart%";
```

COUNT(*)
1 58

SQL 1

```
1 SELECT * FROM tweetsdelivery WHERE text LIKE '%canceled%' ;
```

	id	username	description	location	following	followers	totaltweets	retweetcount	text	hashtags
1	11	NikkiCo04351208	Samoan/native American, 1984 from ...	Seneca, SC	1895	406	1418	0	Wow now I completely understand w...	['canceled', 'Seneca', 'SouthCarolina']
2	19	KristenD0819	MS Warrior öÝši ...	North Carolina, USA	278	130	3244	0	On Friday I placed an order with ...	['Walmart', 'walmart']

Execution finished without errors.
Result: 2 rows returned in 327ms
At line 1:
SELECT * FROM tweetsdelivery WHERE text LIKE '%canceled%' ;

SQL 1

```
1 SELECT COUNT(*) FROM tweetsfart WHERE text LIKE '%walfart%' ;
```

COUNT(*)
1 11

Execution finished without errors.
Result: 1 rows returned in 17ms
At line 1:
SELECT COUNT(*) FROM tweetsfart WHERE text LIKE '%walfart%' ;

SQL 1

```
1 SELECT COUNT (*) FROM tweetsdelivery WHERE text like "%delivery%" ;
2
3
4
```

COUNT (*)
1 20

Execution finished without errors.
Result: 1 rows returned in 18ms
At line 1:
SELECT COUNT (*) FROM tweetsdelivery WHERE text like "%delivery%" ;

```

1 SELECT * FROM tweetsgiftcards WHERE description like "%$750 Walmart gift cards%";
2
3



|   | id | username      | description   | location      | following | followers | totaltweets | retweetcount | text                             | hashtags |
|---|----|---------------|---------------|---------------|-----------|-----------|-------------|--------------|----------------------------------|----------|
| 1 | 2  | walmart_newss | Get \$750 ... | united states | 87        | 21        | 216         | 1            | Walmart \$75... ['walmartgift... |          |
| 2 | 3  | walmart_newss | Get \$750 ... | united states | 87        | 21        | 216         | 1            | Walmart \$75... ['walmartgift... |          |



Execution finished without errors.
Result: 1 rows returned in 18ms
At line 1:
SELECT COUNT (*) FROM tweetsdelivery WHERE text like "%delivery%";


```



```

1 SELECT location , COUNT(*)
2 FROM tweets
3 WHERE description LIKE '%win%' OR description like '%offer%' GROUP BY location;
4
5



| location | COUNT(*) |
|----------|----------|
| 1        | 1        |



Execution finished without errors.
Result: 1 rows returned in 12ms
At line 1:
SELECT location , COUNT(*)
FROM tweets
WHERE description LIKE '%win%' OR description like '%offer%' GROUP BY location;


```



```

1 SELECT *
2 FROM tweets
3 WHERE description LIKE '%win%' OR description like '%offer%' GROUP BY location;
4
5



|   | id | username | description                          | location | following | followers | totaltweets | retweetcount | text                                                     | hashtags | createdat                 |
|---|----|----------|--------------------------------------|----------|-----------|-----------|-------------|--------------|----------------------------------------------------------|----------|---------------------------|
| 1 | 92 | ThisJaxy | Can u shame someone into being a ... |          | 1654      | 367       | 28895       | 348          | I hear people say they never thought ... ['GunViolence'] |          | 2012-12-28 11:48:45+00:00 |



Execution finished without errors.
Result: 1 rows returned in 15ms
At line 1:
SELECT *
FROM tweets
WHERE description LIKE '%win%' OR description like '%offer%' GROUP BY location;


```

```

1 SELECT username, COUNT(totaltweets) FROM tweets ORDER BY totaltweets;
2
3



| username    | COUNT(totaltweets) |
|-------------|--------------------|
| news_cutter | 100                |



Execution finished without errors.
Result: 1 rows returned in 20ms
At line 1:
SELECT username, COUNT(totaltweets) FROM tweets ORDER BY totaltweets;

1 SELECT location, COUNT(*) FROM tweetsfart GROUP BY location;



| location                      | COUNT(*) |
|-------------------------------|----------|
| 1 California sf               | 1        |
| 2 Colorado, USA               | 1        |
| 3 My room                     | 1        |
| 4 NULL                        | 4        |
| 5 Redwood country, California | 1        |
| 6 Sylvania, OH                | 1        |
| 7 Under ur bed, below ur head | 2        |



Execution finished without errors.
Result: 7 rows returned in 19ms
At line 1:
SELECT location, COUNT(*) FROM tweetsfart GROUP BY location;

```

Use cases and creation of views

1. Find total weekly sales of each store.

VIEW --> CREATE VIEW total_sales AS SELECT Store , SUM(Weekly_Sales) as Total_weeklysales
FROM employment GROUP BY Store;

2. Find dates on which Walmart store had Fuel_Price > 3.5 and Holiday_Flag was 0

VIEW --> CREATE VIEW fuelprice AS SELECT Store, Date , Fuel_Price, Holiday_Flag FROM
employment WHERE Holiday_Flag=0 AND Fuel_Price > 3.5;

3. Select min of Unemployment of store 1 and sales between 1542561.09 and 1606629.58

VIEW --> CREATE VIEW Min_unemployemnt AS SELECT MIN(Unemployment), Store from
employment WHERE STORE=1 AND Weekly_Sales BETWEEN 1542561.09 AND 1606629.58
GROUP BY Store;

4. Find average weekly sales of each store.

VIEW --> CREATE VIEW Avg_weekly_sales AS SELECT AVG(Weekly_Sales), Store from
employment GROUP BY Store;

5. List the number of customers in each country. Only include STORES with less than 3 Holiday_flag
VIEW --> CREATE VIEW Sum_holiday_flag AS SELECT Store, SUM(Holiday_Flag) FROM employment GROUP BY Store HAVING SUM(Holiday_Flag) >9;

6. Show all the DaysOnMarket where CITY is Phoenix

VIEW --> CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id where City="Phoenix";

7. Finding out Maximum SizeRank, RegionName, City where DaysOnMarket=106

VIEW --> CREATE VIEW DOM AS SELECT walmart.SizeRank, walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id WHERE SizeRank=(SELECT MAX(SizeRank) from walmart) AND services.DaysOnMarket=106;

8. Finding cities and region where NegativeEquity < Delinquency

VIEW --> CREATE VIEW sales_services AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id WHERE NegativeEquity < Delinquency;

9. Show all cities and regions with any DaysOnMarket they might have

VIEW --> CREATE VIEW All_dom AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart LEFT JOIN services ON walmart.Walmart_id = services.services_id;

10. Show all cities, regions, state with All DaysOnMarket in the table

VIEW --> CREATE VIEW Services_dom AS SELECT walmart.RegionName, walmart.City, walmart.State, services.DaysOnMarket FROM walmart LEFT JOIN services ON walmart.Walmart_id = services.services_id UNION SELECT walmart.RegionName, walmart.City, walmart.State, services.DaysOnMarket FROM walmart LEFT JOIN services ON walmart.Walmart_id = services.services_id ;

11. Show the list of NegativeEquity, Delinquency, regions in Massachusetts state and city is Boston

VIEW --> CREATE VIEW negative_equity AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id WHERE City= "Boston";

12. Finding out Maximum SellForGain, RegionName, City where ZHVI >= 695600

VIEW --> CREATE VIEW ZHVI AS SELECT walmart.SellForGain, walmart.RegionName, walmart.City, services.ZHVI from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id WHERE SellForGain=(SELECT MAX(SellForGain) from walmart) AND services.ZHVI =190900;

13. Show DaysOnMarket in each state

VIEW--> CREATE VIEW DOM_states AS SELECT walmart.State, SUM(services.DaysOnMarket) from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id GROUP BY State;

14. Select store which has the highest weekly sale on this 05-02-2010 date

VIEW --> CREATE VIEW weekly_sales AS SELECT Store, Weekly_Sales, Date from employment where Date = 05-02-2010 AND Weekly_Sales=(SELECT MAX(Weekly_Sales) from employment);

15. Finding out Minimum MarketHealthIndex, RegionName, City where MoM=1.00791936645068

VIEW --> CREATE VIEW min_MHI AS SELECT walmart.MarketHealthIndex, walmart.RegionName, walmart.City, services.MoM from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id WHERE MarketHealthIndex=(SELECT MAX(MarketHealthIndex) from walmart) OR MoM= 1.00791936645068;

SQL Queries and Use cases

1. Find total weekly sales of each store.

SQL--> SELECT Store , SUM(Weekly_Sales) as Total_weeklysales FROM employment GROUP BY Store;

2. Find dates on which Walmart store had Fuel_Price > 3.5 and Holiday_Flag was 0

SQL--> SELECT Store, Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 AND Fuel_Price > 3.5;

3. Select min of Unemployment of store 1 and sales between 1542561.09 and 1606629.58

SQL--> SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales BETWEEN 1542561.09 AND 1606629.58 GROUP BY Store;

4. Find average weekly sales of each store.

SQL--> SELECT AVG(Weekly_Sales), Store from employment GROUP BY Store;

5. List the number of customers in each country. Only include STORES with less than 3 Holiday_flag

SQL--> SELECT Store, SUM(Holiday_Flag) FROM employment GROUP BY Store HAVING SUM(Holiday_Flag) >9;

6. Show all the DaysOnMarket where CITY is Phoenix

SQL--> SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id where City="Phoenix";

7. Finding out Maximum SizeRank, RegionName, City where DaysOnMarket=106

SQL--> SELECT walmart.SizeRank, walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id WHERE SizeRank=(SELECT MAX(SizeRank) from walmart) AND services.DaysOnMarket=106;

8. Finding cities and region where NegativeEquity < Delinquency
SQL--> SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency
from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id WHERE
NegativeEquity < Delinquency;

9. Show all cities and regions with any DaysOnMarket they might have
SQL--> SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart LEFT
JOIN services ON walmart.Walmart_id = services.services_id;

10. Show all cities, regions, state with All DaysOnMarket in the table
SQL--> SELECT walmart.RegionName, walmart.City, walmart.State, services.DaysOnMarket FROM
walmart LEFT JOIN services ON walmart.Walmart_id = services.services_id UNION SELECT
walmart.RegionName, walmart.City, walmart.State, services.DaysOnMarket FROM walmart LEFT JOIN
services ON walmart.Walmart_id = services.services_id ;

11. Show the list of NegativeEquity, Delinquency, regions in Massachusetts state and city is Boston
SQL--> SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency
from walmart INNER JOIN services ON walmart.Walmart_id = services.services_id WHERE
City="Boston";

12. Finding out Maximum SellForGain, RegionName, City where ZHVI >= 695600
SQL--> SELECT walmart.SellForGain, walmart.RegionName, walmart.City, services.ZHVI from
walmart INNER JOIN services ON walmart.Walmart_id = services.services_id WHERE
SellForGain=(SELECT MAX(SellForGain) from walmart) AND services.ZHVI =190900;

13. Show DaysOnMarket in each state
SQL--> SELECT walmart.State, SUM(services.DaysOnMarket) from walmart INNER JOIN services
ON walmart.Walmart_id = services.services_id GROUP BY State;

14. Select store which has the highest weekly sale on this 05-02-2010 date
SQL--> SELECT Store, Weekly_Sales, Date from employment where Date = 05-02-2010 AND
Weekly_Sales=(SELECT MAX(Weekly_Sales) from employment);

15. Finding out Minimum MarketHealthIndex, RegionName, City where MoM=1.00791936645068
SQL--> SELECT walmart.MarketHealthIndex, walmart.RegionName, walmart.City, services.MoM from
walmart INNER JOIN services ON walmart.Walmart_id = services.services_id WHERE
MarketHealthIndex=(SELECT MAX(MarketHealthIndex) from walmart) OR MoM= 1.00791936645068;

16. Create Table Tweets

```
CREATE TABLE tweets(  
id INTEGER PRIMARY KEY AUTOINCREMENT,  
username VARCHAR NOT NULL,  
description VARCHAR NOT NULL,
```

```
location VARCHAR NOT NULL,  
following INTEGER NOT NULL,  
followers INTEGER NOT NULL,  
totaltweets INTEGER NOT NULL,  
retweetcount INTEGER NOT NULL,  
text VARCHAR NOT NULL,  
hastags VARCHAR NOT NULL)
```

17. What are tweets about Walmart delivery ?

```
SQL--> SELECT (*) FROM tweets WHERE description = "walmart delivery";
```

18. How many tweets have there been about #walmartsales?

```
SQL--> SELECT COUNT (*) FROM tweets WHERE hastags ="#walmartsales" ;
```

19. What are the various hashtags related to walmart/sales?

```
SQL--> SELECT COUNT (distinct hastags) FROM tweets WHERE description = "walmartsales";
```

20. How many people/tweets are pleased with Walmart?

```
SQL--> SELECT COUNT(*) FROM tweets WHERE description ="walmart happy customers" OR  
hastags LIKE '%bestbuy%';
```

21. How many tweets about Walmart delivery are there?

```
SQL--> SELECT COUNT(*) FROM tweets WHERE description ="walmartdelivery" OR hastags  
="#walmartdelivery";
```

22. Who are the actors/influencers promoting deals involving Walmart gift cards?

```
SQL--> SELECT * FROM tweets WHERE description = "EmmaWatson" , hastags =  
"#walmartgiftcards" AND description ="influencers" ;
```

23. Which location has received the most tweets?

```
SQL--> SELECT location , COUNT(totaltweets)  
FROM tweets  
GROUP BY location DESC LIMIT=1;
```

24. HP Pavilion product is tweeted the most in which location?

```
SQL--> SELECT location, COUNT(totaltweets) FROM tweets WHERE description like '%HP  
Pavilion%' ORDER BY location LIMIT=1;
```

25. What is the total number of tweets/people upset with Walmart using the hashtag #walfart?

```
SQL--> SELECT COUNT(*) FROM tweets WHERE hastags LIKE '%walfart%';
```

26. What are the most recent Walmart distribution center tweets?

```
SQL--> SELECT * FROM tweets WHERE description = "walmart distribution centre";
```

27. How many customers are moving from walmart to target?

SQL--> SELECT COUNT(*) FROM tweets WHERE description ="no walmart yes target";

28. How many customers are irritated when their order is lost or delivered late?

SQL--> SELECT COUNT(*) FROM tweets WHERE description LIKE '%latedelivery%' OR description LIKE '%lost%' AND hastags LIKE '%walfart%';

29. Which top 3 cities received the most gift cards?

SQL--> SELECT location , COUNT(*)
FROM tweets
WHERE description LIKE '%win%' OR description like '%offer%' GROUP BY location LIMIT=3;

30. Which of the following people are top 3 Walmart customer who frequently tweet about Walmart?

SQL--> SELECT username, COUNT(totaltweets) FROM tweets ORDER BY totaltweets LIMIT=3;

Twitter Code

```
# Python Script to Extract tweets of a
# particular Hashtag using Tweepy and Pandas

# import modules
import pandas as pd
import tweepy

# function to display data of each tweet
def printtweetdata(n, ith_tweet):
    print()
    print(f"Tweet {n}:")
    print(f"Username:{ith_tweet[0]}")
    print(f"Description:{ith_tweet[1]}")
    print(f"Location:{ith_tweet[2]}")
    print(f"Following Count:{ith_tweet[3]}")
    print(f"Follower Count:{ith_tweet[4]}")
    print(f"Total Tweets:{ith_tweet[5]}")
    print(f"Retweet Count:{ith_tweet[6]}")
    print(f"Tweet Text:{ith_tweet[7]}")
    print(f"Hashtags Used:{ith_tweet[8]}")
    print(f"Created At:{ith_tweet[9]}")

# function to perform data extraction
def scrape(words, date_since, numtweets):

    # Creating DataFrame using pandas
```

```

db = pd.DataFrame(columns=['username',
                           'description',
                           'location',
                           'following',
                           'followers',
                           'totaltweets',
                           'retweetcount',
                           'text',
                           'hashtags',
                           'createdat'])

# We are using .Cursor() to search
# through twitter for the required tweets.
# The number of tweets can be
# restricted using .items(number of tweets)
tweets = tweepy.Cursor(api.search_tweets,
                       words, lang="en",
                       since_id=date_since,
                       tweet_mode='extended').items(numtweet)

# .Cursor() returns an iterable object. Each item in
# the iterator has various attributes
# that you can access to
# get information about each tweet
list_tweets = [tweet for tweet in tweets]

# Counter to maintain Tweet Count
i = 1

# we will iterate over each tweet in the
# list for extracting information about each tweet
for tweet in list_tweets:
    username = tweet.user.screen_name
    description = tweet.user.description
    location = tweet.user.location
    following = tweet.user.friends_count
    followers = tweet.user.followers_count
    totaltweets = tweet.user.statuses_count
    retweetcount = tweet.retweet_count
    hashtags = tweet.entities['hashtags']
    createdat = tweet.user.created_at

    # Retweets can be distinguished by

```

```

# a retweeted_status attribute,
# in case it is an invalid reference,
# except block will be executed
try:
    text = tweet.retweeted_status.full_text
except AttributeError:
    text = tweet.full_text
hashtext = list()
for j in range(0, len(hashtags)):
    hashtext.append(hashtags[j]['text'])

# Here we are appending all the
# extracted information in the DataFrame
ith_tweet = [username, description,
             location, following,
             followers, totaltweets,
             retweetcount, text, hashtext, createdat]
db.loc[len(db)] = ith_tweet

# Function call to print tweet data on screen
printtweetdata(i, ith_tweet)
i = i+1
filename = 'GFG_tweets.csv'

# we will save our database as a CSV file.
db.to_csv(filename)

if __name__ == '__main__':
    # Enter your own credentials obtained
    # from your developer account
    consumer_key = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
    consumer_secret = "XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
    access_key =
    "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
    access_secret =
    "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

    auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_key, access_secret)
    api = tweepy.API(auth)

    # Enter Hashtag and initial date

```

```

print("Enter Twitter HashTag to search for")
words = input()
print("Enter Date since The Tweets are required in yyyy-mm--dd")
date_since = input()

# number of tweets you want to extract in one run
numtweet = 100
scrape(words, date_since, numtweet)
print('Scraping has completed!')

```

Updating Database from CSV

```

# Import required modules
import csv
import sqlite3

# Connecting to the geeks database
connection = sqlite3.connect('Tweets_Extracted.db')

# Creating a cursor object to execute
# SQL queries on a database table
cursor = connection.cursor()

# Table Definition
create_table = "CREATE TABLE tweets(
                    id INTEGER PRIMARY KEY AUTOINCREMENT,
                    username VARCHAR NOT NULL,
                    description VARCHAR NOT NULL,
                    location VARCHAR NOT NULL,
                    following INTEGER NOT NULL,
                    followers INTEGER NOT NULL,
                    totaltweets INTEGER NOT NULL,
                    retweetcount INTEGER NOT NULL,
                    text VARCHAR NOT NULL,
                    hastags VARCHAR NOT NULL
                );
            """

# Creating the table into our
# database
cursor.execute(create_table)

# Opening the tweets-records.csv file

```

```

file = open('GFG_tweets.csv' , errors='ignore')

# Reading the contents of the
# tweets-records.csv file
contents = csv.reader(file)

# SQL query to insert data into the
# tweets table
# Similar Line is used remaining tables.
insert_records = "INSERT INTO tweets (username, description, location, following, followers,
totaltweets, retweetcount, text, hastags) VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?)"

# Importing the contents of the file
# into our tweets table
cursor.executemany(insert_records, contents)

# SQL query to retrieve all data from
# the person table To verify that the
# data of the csv file has been successfully
# inserted into the table
select_all = "SELECT * FROM tweets"
rows = cursor.execute(select_all).fetchall()

# Output to the console screen
for r in rows:
    print(r)

# Committing the changes
connection.commit()

# closing the database connection
connection.close()

```

Python Code for Data Cleaning , Munging

```

import pandas as pd
import numpy as np

#import matplotlib.pyplot as plt

#Dataset 1
walmart_path = 'D:\DMDD\Assignment 3\myCity.csv'

```

```

walmart_ori = pd.read_csv(walmart_path)
walmart = walmart_ori.copy()

walmart.head()

walmart.info()

#Function to fetch missing values from Dataset 1
def missing_cols(walmart):
    """prints out columns with its amount of missing values"""
    total = 0
    for col in walmart.columns:
        missing_vals = walmart[col].isnull().sum()
        total += missing_vals
        if missing_vals != 0:
            print(f"{col} => {walmart[col].isnull().sum()}")

    if total == 0:
        print("no missing values left")

#Missing Columns Values in Walmart Dataframe
missing_cols(walmart)

def perc_missing(walmart):
    """prints out columns with missing values with its %"""
    for col in walmart.columns:
        pct = walmart[col].isna().mean() * 100
        if (pct != 0):
            print(f'{col} => {pct}%'.format(col, round(pct, 2)))

#Percentage Wise Missing Values
perc_missing(walmart)

# Drop unnecessary columns that are not important
colsToDelete = ['StockOfREOs','PrevForeclosed','ForeclosureRatio']

walmart.drop(colsToDelete, axis=1, inplace=True)

#Missing Values Redefined
missing_cols(walmart)

#Percentage Wise Missing Values
perc_missing(walmart)

```

```

# imputing with bfill or ffill
walmart['Metro'].bfill(inplace=True)
walmart['Metro'].ffill(inplace=True)
walmart['SizeRank'].bfill(inplace=True)
walmart['SizeRank'].ffill(inplace=True)
walmart['SellForGain'].bfill(inplace=True)
walmart['SellForGain'].ffill(inplace=True)
walmart['ZHVI'].bfill(inplace=True)
walmart['ZHVI'].ffill(inplace=True)
walmart['MoM'].bfill(inplace=True)
walmart['MoM'].ffill(inplace=True)
walmart['ForecastYoYPctChange'].ffill(inplace=True)
walmart['ForecastYoYPctChange'].bfill(inplace=True)
walmart['YoY'].ffill(inplace=True)
walmart['YoY'].ffill(inplace=True)
walmart['Delinquency'].ffill(inplace=True)
walmart['Delinquency'].ffill(inplace=True)
walmart['DaysOnMarket'].ffill(inplace=True)
walmart['DaysOnMarket'].ffill(inplace=True)
walmart['NegativeEquity'].ffill(inplace=True)
walmart['NegativeEquity'].ffill(inplace=True)

#DF Post Cleaning
missing_cols(walmart)

#Cleaned DF 1
print (walmart)

#Cleaned Data Inserted into CSV
csv_data = walmart.to_csv('D:\DMDD\Assignment 3\Cleaned_1.csv', index = False)

# Import required modules
import csv
import sqlite3

# Connecting to the geeks database
connection = sqlite3.connect('D:\DMDD\Assignment 3\Cleaned_DB.db')

# Creating a cursor object to execute
# SQL queries on a database table

```

```

cursor = connection.cursor()

# Table Definition
create_table = "CREATE TABLE IF NOT EXISTS walmart (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                RegionType VARCHAR NOT NULL,
                RegionName VARCHAR NOT NULL,
                City VARCHAR NOT NULL,
                State VARCHAR NOT NULL,
                Metro VARCHAR NOT NULL,
                SizeRank INTEGER NOT NULL,
                MarketHealthIndex INTEGER NOT NULL,
                SellForGain INTEGER NOT NULL,
                ZHVI INTEGER NOT NULL,
                MoM INTEGER NOT NULL,
                YoY INTEGER NOT NULL,
                ForecastYoYPctChange INTEGER NOT NULL,
                NegativeEquity INTEGER NOT NULL,
                Delinquency INTEGER NOT NULL,
                DaysOnMarket INTEGER NOT NULL
            );
            """

# Creating the table into our
# database
cursor.execute(create_table)

# Opening the tweets-records.csv file
file = open('D:\\DMDD\\Assignment 3\\Cleaned.csv', errors='ignore')

# Reading the contents of the
# tweets-records.csv file
contents = csv.reader(file)

# SQL query to insert data into the
# tweets table
insert_records = "INSERT INTO walmart (RegionType, RegionName, City, State, Metro,
                                         SizeRank, MarketHealthIndex, SellForGain, ZHVI, MoM, YoY,
                                         ForecastYoYPctChange, NegativeEquity, Delinquency, DaysOnMarket) VALUES( ?, ?, ?, ?, ?, ?,
                                         ?, ?, ?, ?, ?, ?, ?, ?, ?)"

# Importing the contents of the file

```

```

# into our tweets table
cursor.executemany(insert_records, contents)

# SQL query to retrieve all data from
# the person table To verify that the
# data of the csv file has been successfully
# inserted into the table
# Change Table to tweets after the SQL
select_all = "SELECT * FROM walmart"
rows = cursor.execute(select_all).fetchall()

# Output to the console screen
for r in rows:
    print(r)

# Committing the changes
connection.commit()

# closing the database connection
connection.close()

#Dataset 2
walmart_path_1 = 'D:\DMDD\Assignment 3\Walmart.csv'

walmart_ori_1 = pd.read_csv(walmart_path_1)
walmart_1 = walmart_ori_1.copy()

walmart_1.head()

walmart_1.info()

#Function to fetch missing values from Dataset 1
def missing_cols(walmart_1):
    "prints out columns with its amount of missing values"
    total = 0
    for col in walmart_1.columns:
        missing_vals = walmart_1[col].isnull().sum()
        total += missing_vals
        if missing_vals != 0:
            print(f"\n{col} => {walmart_1[col].isnull().sum()}\n")
    if total == 0:

```

```

print("no missing values left")

missing_cols(walmart_1)

# Connecting to the geeks database
connection = sqlite3.connect('D:\DMDD\Assignment 3\Cleaned_DB.db')

# Creating a cursor object to execute
# SQL queries on a database table
cursor = connection.cursor()

# Table Definition
create_table = """CREATE TABLE IF NOT EXISTS employment (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    Store INTEGER NOT NULL,
    Date INTEGER NOT NULL,
    Weekly_Sales INTEGER NOT NULL,
    Holiday_Flag INTEGER NOT NULL,
    Temperature INTEGER NOT NULL,
    Fuel_Price INTEGER NOT NULL,
    CPI INTEGER NOT NULL,
    Unemployment INTEGER NOT NULL
);
"""

# Creating the table into our
# database
cursor.execute(create_table)

# Opening the tweets-records.csv file
file = open('D:\DMDD\Assignment 3\Walmart.csv' , errors='ignore')

# Reading the contents of the
# tweets-records.csv file
contents = csv.reader(file)

# SQL query to insert data into the
# tweets table
insert_records = "INSERT INTO employment (Store, Date, Weekly_Sales, Holiday_Flag, Temperature, Fuel_Price, CPI, Unemployment) VALUES( ?, ?, ?, ?, ?, ?, ?, ?)"

# Importing the contents of the file

```

```

# into our tweets table
cursor.executemany(insert_records, contents)

# SQL query to retrieve all data from
# the person table To verify that the
# data of the csv file has been successfully
# inserted into the table
# Change Table to tweets after the SQL
select_all = "SELECT * FROM employment"
rows = cursor.execute(select_all).fetchall()

# Output to the console screen
for r in rows:
    print(r)

# Committing the changes
connection.commit()

# closing the database connection
connection.close()

```

Use Cases:

- Which retailer has the highest sales?
- Which retailer's standard deviation is the highest
- Consequently, there are wide variations in sales. Likewise, determine the ratio of the mean to the standard deviation.
- Which retailer(s) has/have the best quarterly growth rate in Q3?
- Sales may suffer over certain holidays. Find out which holidays have more sales than the average non-holiday season sales for all of the stores combined.
- Give insights and a monthly and semester snapshot of the sales in units.
- Which of the retailers is closest to public transportation and makes commuting easier?
- What other stores are located near the Walmart retailers? Because Walmart may lose some customers if they are drawn to other stores such as Target and Giant...
- A thorough summary of the sales analysis so that other market competitors or new startups can identify the elements that boost sales and the difficulties that depress sales.
- What is the category in which Walmart sees the most customer traffic?
- Which retailer buys products from the farmers market?
- What is the most cost-effective logistics and shipping organization that can be considered in order to fulfill the restocking of goods?
- What's the attrition rate of employees in the Walmart retailer?
- What is the timetable / ETA for retailers to restock supplies?

- How long does the billing line typically wait?
- What standard has Walmart set for employee education while hiring?
- What is the typical employee happiness index based on employment?
- Does a retailer really work upon the customer feedback?
- What's the least time taken by the delivery vendors to ship the packages to the desired address?
- Do the retailers ship the package outside their jurisdiction? If not, why?
- What are the trivial requirements to get a membership of Walmart shops? Is SSN blocked to get a membership? If yes, how can it be avoided? As it might lose potential customers like students.
- Are the groceries made available with dietary allergies and other types of customers in mind? (Kosher, Vegan, Lactose Intolerant)
- Is the store easily accessible to people with disabilities, the elderly, and pregnant women?
- Do retailers develop marketing strategies?
- What percentage of goods expire or are thrown away because they are not purchased?
- What products do Walmart retailers lack? Why doesn't the retailer stock that genre/category?
- How many competitive retailer markets are available in the store's immediate vicinity? Does this have an impact on the store's sales?
- How many customers prefer to shop online rather than in the store?
- How frequently does the retailer need to advertise the store?
- How much of an impact does the advertisement have on sales?
- Is the retailer in a prime location in the city?
- Is there parking available at the specific retailer store?
- Is self-checkout available at the store?

Steps performed to get desired database quality:

- Investigated datasets from kaggle and other legitimate data source platforms.
- Verified the datasets' authenticity.
- Created some Python scripts to convert csv data to database and insert it into appropriate tables.
- Assign key constraints to multiple tables in order to perform useful operations.
- Executed SQL statements to retrieve information as defined by usecases.
- Cleaned up the database and removed inconsistencies using Python scripts.
- Created an ER and UML diagram to comprehend the flow and database skeleton.
- Instead of using regular SQL select statements, VIEWS for SQL were created to simplify data retrieval.
- Normalized the data using 1NF, 2NF, 3NF, and 4NF normalization methods.

Outputs of all 15 views

The screenshot shows the MySQL Workbench interface with two connections: "Local instance 3306" and "Local instance 3306". The "Schemas" tab is selected, showing the "walmart" schema expanded. The "Tables" node under "walmart" contains 15 entries, all of which are marked as "VIEW". The "Query" tab displays the SQL command used to generate the results:

```
1 • use walmart;
2 • SHOW FULL TABLES WHERE table_type = 'VIEW';
3
```

The "Result Grid" pane shows the following data:

Tables_in_walmart	Table_type
all_dom	VIEW
avg_weekly_sales	VIEW
dom	VIEW
dom_phoenix	VIEW
dom_states	VIEW
fuelprice	VIEW
min_mhi	VIEW
min_unemployment	VIEW
negative_equity	VIEW
sales_services	VIEW
services_dom	VIEW
sum_holiday_flag	VIEW
total_sales	VIEW
weekly_sales	VIEW
zhvi	VIEW

The "Session" tab at the bottom shows the following activity:

Action Output	Time	Action	Response	Duration / Fetch
1	18:22:48	use walmart	0 row(s) affected	0.00030 sec
2	18:22:48	SHOW FULL TABLES WHERE table_type = 'VIEW'	15 row(s) returned	0.0051 sec / 0

MySQL Workbench

Administration Schemas Query 1 create-databases

SCHEMAS Filter objects

- sql_hr
- sql_inventory
- sql_invoicing
- sql_store
- sys
- Walmart**

Tables

- employment
- services
- walmart

Views

Stored Procedures

Functions

Object Info Session Schema: Walmart

Action Output

```

1 USE Walmart;
2 CREATE VIEW fuelprice AS SELECT Store, Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 AND Fuel_Price > 3.5;
3 SELECT * FROM fuelprice;
4
5

```

Result Grid Filter Rows: Search Export: Fetch rows:

Store	Date	Fuel_Price	Holiday_Flag
1	01-04-2011	3.524	0
1	08-04-2011	3.622	0
1	15-04-2011	3.743	0
1	22-04-2011	3.648	0
1	29-04-2011	3.81	0
1	06-05-2011	3.906	0
1	13-05-2011	3.899	0
1	20-05-2011	3.907	0
1	27-05-2011	3.786	0
1	03-06-2011	3.699	0
1	10-06-2011	3.648	0
1	17-06-2011	3.67	0
1	24-06-2011	3.694	0
1	01-07-2011	3.524	0
1	15-07-2011	3.575	0
1	22-07-2011	3.651	0
1	29-07-2011	3.682	0
1	05-08-2011	3.684	0
1	12-08-2011	3.638	0

fuelprice 2

Object Info Session Schema: Walmart

Action Output

Time	Action	Response	Duration / Fetch Time
14:00:02	PREPARE SP1 FOR INSERT INTO `Walmart`.`employment`(`Store`, `Date`, `Weekly_Sales`, `Holiday_Flag`, `Temperature`, `Fuel_Price`)	OK	0.000 sec
18 14:05:08	DEALLOCATE PREPARE stmt	OK	0.000 sec
19 14:09:35	CREATE VIEW store_name AS SELECT Store , SUM(Weekly_Sales) FROM employment GROUP BY Store	0 row(s) affected	0.0082 sec
20 10:13:04	USE Walmart	0 row(s) affected	0.0011 sec
21 10:13:06	USE Walmart	0 row(s) affected	0.00035 sec
22 18:42:55	USE Walmart	0 row(s) affected	0.0033 sec
23 18:42:55	CREATE VIEW total_sales AS SELECT Store , SUM(Weekly_Sales) as Total_Weeklysales FROM employment GROUP BY Store	0 row(s) affected	0.0031 sec
24 18:44:37	SELECT * FROM total_sales LIMIT 0, 1000	45 row(s) returned	0.0087 sec / 0.00001...
25 18:47:44	CREATE VIEW fuelprice AS SELECT Store, Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 AND Fuel_Price > 3.5;	0 row(s) affected	0.0020 sec
26 18:47:44	SELECT * FROM fuelprice	1000 row(s) returned	0.0037 sec / 0.00091...
26 18:47:49	SELECT * FROM fuelprice LIMIT 0, 1000	0 row(s) affected	0.0028 sec
27 18:49:02	CREATE VIEW min_unemployment AS SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales BETWEEN 1542561.09 AND 1606629.58 GROUP BY Store	1 row(s) returned	0.0066 sec / 0.0000...
28 18:49:04	SELECT * FROM min_unemployment	1 row(s) returned	0.0037 sec / 0.00091...

Query Completed

MySQL Workbench

Administration Schemas Query 1 create-databases

SCHEMAS Filter objects

- sql_hr
- sql_inventory
- sql_invoicing
- sql_store
- sys
- Walmart**

Tables

- employment
- services
- walmart

Views

Stored Procedures

Functions

Object Info Session Schema: Walmart

Action Output

```

1 USE Walmart;
2 CREATE VIEW Min_unemployment AS SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales BETWEEN 1542561.09 AND 1606629.58 GROUP BY Store;
3 SELECT * FROM Min_unemployment;
4
5

```

Result Grid Filter Rows: Search Export: Fetch rows:

MIN(Unemployment)	Store
6.573	1

Object Info Session Schema: Walmart

Action Output

Time	Action	Response	Duration / Fetch Time
19 14:09:30	CREATE VIEW store_name AS SELECT Store , SUM(Weekly_Sales) AS Total_Weeklysales FROM employment GROUP BY Store	0 row(s) affected	0.0002 sec
20 10:13:04	USE Walmart	0 row(s) affected	0.0011 sec
21 10:13:06	USE Walmart	0 row(s) affected	0.00035 sec
22 18:42:55	USE Walmart	0 row(s) affected	0.0033 sec
23 18:42:55	CREATE VIEW total_sales AS SELECT Store , SUM(Weekly_Sales) AS Total_Weeklysales FROM employment GROUP BY Store	0 row(s) affected	0.0031 sec
24 18:44:37	SELECT * FROM total_sales LIMIT 0, 1000	45 row(s) returned	0.0087 sec / 0.00001...
25 18:47:44	CREATE VIEW fuelprice AS SELECT Store, Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 AND Fuel_Price > 3.5;	0 row(s) affected	0.0020 sec
26 18:47:49	SELECT * FROM fuelprice	1000 row(s) returned	0.0037 sec / 0.00091...
27 18:49:02	CREATE VIEW min_unemployment AS SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales BETWEEN 1542561.09 AND 1606629.58 GROUP BY Store	0 row(s) affected	0.0028 sec
28 18:50:04	SELECT * FROM min_unemployment	1 row(s) returned	0.0066 sec / 0.0000...

Query Completed

MySQL Workbench

Administration Schemas Query 1 create-databases Limit to 1000 rows

SCHEMAS

Filter objects

- > sql_hr
- > sql_inventory
- > sql_invoicing
- > sql_store
- > sys
- Walmart**
 - Tables
 - > employment
 - > services
 - > walmart
 - Views
 - Stored Procedures
 - Functions

100% 32.3 | Result Grid Filter Rows: Search Export:

```

1 • USE Walmart;
2 • CREATE VIEW Avg_weekly_sales AS SELECT AVG(Weekly_Sales), Store FROM employment GROUP BY Store;
3 • SELECT * FROM Avg_weekly_sales;
4
5
6

```

Result Grid Filter Rows: Search Export:

Avg(Weekly_Sales)	Store
1555264.3975524479	1
1925751.3355244761	2
402704.44104895106	3
209400.6962000000	4
318011.8104895105	5
1564728.186933063	6
570617.3096713287	7
908749.5183916084	8
543980.5523776223	9
1899424.5726753742	10
136200.9944755238	11
1009001.699999999	12
2003620.286293707	13
2020679.400979021	14
623312.4749550348	15
519247.7300699302	16
893581.390419581	17
1084718.421048951	18
1444999.0356843356	19

Object Info Session Avg_weekly_sales 4 Read Only

Schema: Walmart

Action Output

Time	Action	Response	Duration / Fetch Time
11:12:00	USE Walmart;	0 row(s) affected	0.0033 sec
18:42:55	USE Walmart	0 row(s) affected	0.0031 sec
18:42:55	CREATE VIEW total_sales AS SELECT Store , SUM(Weekly_Sales) as Total_weeklysales FROM employment GROUP BY Store	45 row(s) returned	0.0087 sec / 0.00001...
18:44:37	SELECT * FROM total_sales LIMIT 0, 1000	0 row(s) affected	0.0020 sec
18:47:44	CREATE VIEW fuelprice AS SELECT Store, Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 AND Fuel_Pri...	1000 row(s) returned	0.0037 sec / 0.00091...
18:47:49	SELECT * FROM fuelprice LIMIT 0, 1000	0 row(s) affected	0.0028 sec
18:49:02	CREATE VIEW Min_unemployment AS SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales B...	1 row(s) returned	0.0066 sec / 0.0000...
18:50:04	SELECT * FROM Min_unemployment LIMIT 0, 1000	0 row(s) affected	0.0018 sec
18:50:50	CREATE VIEW Avg_weekly_sales AS SELECT AVG(Weekly_Sales), Store from employment GROUP BY Store	45 row(s) returned	0.011 sec / 0.000012...
18:51:17	SELECT * FROM Avg_weekly_sales LIMIT 0, 1000	0 row(s) affected	0.0022 sec
18:53:09	CREATE View Sum_holiday_flag AS SELECT Store, SUM(Holiday_Flag) FROM employment GROUP BY Store HAVING SUM(Holiday_F...	45 row(s) returned	0.011 sec / 0.000010...
18:53:14	SELECT * FROM Sum_holiday_flag LIMIT 0, 1000	0 row(s) affected	0.0022 sec

Query Completed

MySQL Workbench

Administration Schemas Query 1 create-databases Limit to 1000 rows

SCHEMAS

Filter objects

- > sql_hr
- > sql_inventory
- > sql_invoicing
- > sql_store
- > sys
- Walmart**
 - Tables
 - > employment
 - > services
 - > walmart
 - Views
 - Stored Procedures
 - Functions

100% 26.3 | Result Grid Filter Rows: Search Export:

```

1 • USE Walmart;
2 • CREATE VIEW Sum_holiday_flag AS SELECT Store, SUM(Holiday_Flag) FROM employment GROUP BY Store HAVING SUM(Holiday_Flag) >0;
3 • SELECT * FROM Sum_holiday_flag;
4
5
6

```

Result Grid Filter Rows: Search Export:

Store	SUM(Holiday_Flag)
1	10
2	10
3	10
4	10
5	10
6	10
7	10
8	10
9	10
10	10
11	10
12	10
13	10
14	10
15	10
16	10
17	10
18	10
19	10
20	10
21	10
22	10
23	10
24	10
25	10
26	10
27	10
28	10
29	10
30	10
31	10
32	10

Object Info Session Sum_holiday_flag 5 Read Only

Schema: Walmart

Action Output

Time	Action	Response	Duration / Fetch Time
10:42:00	CREATE VIEW total_sales AS SELECT Store , SUM(Weekly_Sales) as Total_weeklysales FROM employment GROUP BY Store	0 row(s) affected	0.0033 sec
18:44:37	SELECT * FROM total_sales LIMIT 0, 1000	45 row(s) returned	0.0087 sec / 0.00001...
18:47:44	CREATE VIEW fuelprice AS SELECT Store, Date , Fuel_Price, Holiday_Flag FROM employment WHERE Holiday_Flag=0 AND Fuel_Pri...	0 row(s) affected	0.0020 sec
18:47:49	SELECT * FROM fuelprice LIMIT 0, 1000	1000 row(s) returned	0.0037 sec / 0.00091...
18:49:02	CREATE VIEW Min_unemployment AS SELECT MIN(Unemployment), Store from employment WHERE STORE=1 AND Weekly_Sales B...	0 row(s) affected	0.0028 sec
18:50:04	SELECT * FROM Min_unemployment LIMIT 0, 1000	1 row(s) returned	0.0066 sec / 0.0000...
18:50:50	CREATE VIEW Avg_weekly_sales AS SELECT AVG(Weekly_Sales), Store from employment GROUP BY Store	0 row(s) affected	0.0018 sec
18:51:17	SELECT * FROM Avg_weekly_sales LIMIT 0, 1000	45 row(s) returned	0.011 sec / 0.000012...
18:53:09	CREATE View Sum_holiday_flag AS SELECT Store, SUM(Holiday_Flag) FROM employment GROUP BY Store HAVING SUM(Holiday_F...	0 row(s) affected	0.0022 sec
18:53:14	SELECT * FROM Sum_holiday_flag LIMIT 0, 1000	45 row(s) returned	0.011 sec / 0.000010...

Query Completed

Local Instance 3306

Administration Schemas Query 1 create-databases

SCHEMAS

Filter objects

- sql_hr
- sql_inventory
- sql_invoicing
- sql_store
- sys
- Walmart**
 - Tables
 - employment
 - services**
 - walmart
 - Views
 - Stored Procedures
 - Functions

Query 1 Limit to 1000 rows

```

1 • USE Walmart;
2 • CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON walmart.Walmart_id = services
3 • SELECT * FROM DOM_Phoenix;
4
5
6

```

Result Grid Filter Rows: Search Export:

RegionName	City	DaysOnMarket
Phoenix	Phoenix	81
Phoenix	Phoenix	120
Phoenix	Phoenix	139

Object Info Session DOM_Phoenix 6 Read Only

Table: services

Action Output

Time	Action	Response	Duration / Fetch Time
10:00:09	SELECT * FROM Walmart LIMIT 0, 1000	1000 row(s) returned	0.0011 sec / 0.0029 sec...
19:01:22	Apply changes to walmart	Error 1171: All parts of a PRIMARY KEY must be NOT...	
19:03:23	SELECT * FROM Walmart.walmart LIMIT 0, 1000	1000 row(s) returned	0.0011 sec / 0.0029 sec...
19:03:42	Apply changes to walmart	Error 1171: All parts of a PRIMARY KEY must be NOT...	
19:05:44	Apply changes to walmart	Changes applied	
19:06:21	CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN serv...	Error Code: 1054. Unknown column 'services.services...' 0.00094 sec	
19:06:32	SELECT * FROM Walmart.services LIMIT 0, 1000	1000 row(s) returned	0.0023 sec / 0.0071 sec...
19:07:49	Apply changes to services	Changes applied	
19:08:59	CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN serv...	0 row(s) affected	0.0019 sec
19:09:26	SELECT * FROM DOM_Phoenix LIMIT 0, 1000	3 row(s) returned	0.012 sec / 0.000012...
19:09:26	SELECT * FROM DOM LIMIT 0, 1000	0 row(s) returned	0.0024 sec
19:10:18	CREATE VIEW DOM AS SELECT walmart.SizeRank, walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER...	0 row(s) affected	0.014 sec / 0.000012...
19:10:52	SELECT * FROM DOM LIMIT 0, 1000	0 row(s) returned	

Query Completed

Local Instance 3306

Administration Schemas Query 1 create-databases

SCHEMAS

Filter objects

- sql_hr
- sql_inventory
- sql_invoicing
- sql_store
- sys
- Walmart**
 - Tables
 - employment
 - services**
 - walmart
 - Views
 - Stored Procedures
 - Functions

Query 1 Limit to 1000 rows

```

1 • USE Walmart;
2 • CREATE VIEW DOM AS SELECT walmart.SizeRank, walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON walmart.Walmart_id = services
3 • SELECT * FROM DOM;
4
5
6

```

Result Grid Filter Rows: Search Export:

SizeRank	RegionName	City	DaysOnMarket
----------	------------	------	--------------

Object Info Session DOM 7 Read Only

Table: services

Action Output

Time	Action	Response	Duration / Fetch Time
19:03:29	SELECT * FROM Walmart LIMIT 0, 1000	1000 row(s) returned	0.0011 sec / 0.0029 sec...
19:03:42	Apply changes to walmart	Error 1171: All parts of a PRIMARY KEY must be NOT...	
19:05:44	Apply changes to walmart	Changes applied	
19:06:21	CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN serv...	Error Code: 1054. Unknown column 'services.services...' 0.00094 sec	
19:06:32	SELECT * FROM Walmart.services LIMIT 0, 1000	1000 row(s) returned	0.0023 sec / 0.0071 sec...
19:07:49	Apply changes to services	Changes applied	
19:08:59	CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN serv...	0 row(s) affected	0.0019 sec
19:09:26	SELECT * FROM DOM LIMIT 0, 1000	3 row(s) returned	0.012 sec / 0.000012...
19:10:18	CREATE VIEW DOM AS SELECT walmart.SizeRank, walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER...	0 row(s) affected	0.0024 sec
19:10:52	SELECT * FROM DOM LIMIT 0, 1000	0 row(s) returned	

Query Completed

Local Instance 3306

Administration Schemas Query 1 create-databases Limit to 1000 rows

SCHEMAS

Filter objects

- > sql_hr
- > sql_inventory
- > sql_invoicing
- > sql_store
- > sys
- Walmart
 - Tables
 - > employment
 - > services
 - walmart
 - Views
 - Stored Procedures
 - Functions

Query 1

```

1 • USE Walmart;
2 • CREATE VIEW sales_services AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency FROM walmart INNER JOIN services ON wa
3 • SELECT * FROM sales_services;
4
5

```

Result Grid

RegionName	City	NegativeEquity	Delinquency
New York	New York	0.067887526157941	0.099894063036363
Phoenix	Phoenix	0.076207912053059	0.115986084656415
Salt Lake City	Salt Lake City	0.071328932282443	0.1407901478071128
Houston	Houston	0.069423535984091	0.106571195176699
Vancouver	Vancouver	0.060943535984091	0.080572111231384
Queen Creek	Queen Creek	0.052721159156326	0.1143676484369
Coral Springs	Coral Springs	0.099032927123113	0.0996711287744
Concord	Concord	0.015111253885151	0.027048093303679
Lakewood Township	Lakewood Township	0.122914693982722	0.13002212261005
Livermore	Livermore	0.07229073998967	0.085440021496826
Concord	Concord	0.07229073998967	0.085440021496826
Nashua	Nashua	0.0572457226909	0.0911783785177
Chapel Hill	Chapel Hill	0.05846176158471	0.10468971155397
Covina	Covina	0.029568893380226	0.147600790947179
New Britain	New Britain	0.046392631758485	0.055147058823529
Loveland	Loveland	0.072025565383937	0.085324232081911
Cranston	Cranston	0.05706036232988	0.06135874350542
Framingham	Framingham	0.053925921596679	0.070930465573983
Conyers	Conyers	0.11462279382428	0.130784491765878

Object Info Session sales_services 8

Table: services

Action Output

Time	Action	Response	Duration / Fetch Time
19:00:44	Apply changes to services	Changes applied	
41 19:06:21	CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON wa...	Error Code: 1054. Unknown column 'services.services...' in...	0.00094 sec
42 19:06:32	SELECT * FROM Walmart.services LIMIT 0, 1000	1000 row(s) returned	0.0023 sec / 0.0071 s...
43 19:06:32	Changes applied		
44 19:07:49	Apply changes to services	0 row(s) affected	0.0019 sec
45 19:08:59	CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON wa...	3 row(s) affected	0.012 sec / 0.000011...
46 19:09:26	SELECT * FROM DOM_Phoenix LIMIT 0, 1000	0 row(s) affected	0.0024 sec
47 19:10:18	CREATE VIEW DOM_AS SELECT walmart.SizeRank, walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER...	0 row(s) returned	0.014 sec / 0.000012...
48 19:10:52	SELECT * FROM DOM LIMIT 0, 1000	0 row(s) affected	0.0017 sec
49 19:12:01	CREATE VIEW sales_services AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from ...	0 row(s) affected	
50 19:12:21	SELECT * FROM sales_services LIMIT 0, 1000	1000 row(s) returned	0.0042 sec / 0.0060...

Object Info Session All_dom 9

Table: services

Action Output

Time	Action	Response	Duration / Fetch Time
19:00:32	SELECT * FROM Walmart.services LIMIT 0, 1000	1000 row(s) returned	0.0023 sec / 0.0071 s...
43 19:07:49	Apply changes to services	Changes applied	
45 19:08:59	CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON wa...	Error Code: 1054. Unknown column 'services.services...' in...	0.00094 sec
46 19:09:26	SELECT * FROM DOM_Phoenix LIMIT 0, 1000	3 row(s) returned	0.012 sec / 0.000011...
47 19:10:18	CREATE VIEW DOM_AS SELECT walmart.SizeRank, walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER...	0 row(s) affected	0.0024 sec
48 19:10:52	SELECT * FROM DOM LIMIT 0, 1000	0 row(s) returned	0.014 sec / 0.000012...
49 19:12:01	CREATE VIEW sales_services AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from ...	1000 row(s) returned	0.0017 sec
50 19:12:21	SELECT * FROM sales_services LIMIT 0, 1000	0 row(s) affected	0.0042 sec / 0.0060...
51 19:13:19	CREATE VIEW All_dom AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart LEFT JOIN services...	1000 row(s) returned	0.0027 sec
52 19:13:43	SELECT * FROM All_dom LIMIT 0, 1000	1000 row(s) returned	0.0071 sec / 0.0075...

Query Completed

Local Instance 3306

Administration Schemas Query 1 create-databases Limit to 1000 rows

SCHEMAS

Filter objects

- > sql_hr
- > sql_inventory
- > sql_invoicing
- > sql_store
- > sys
- Walmart
 - Tables
 - > employment
 - > services
 - walmart
 - Views
 - Stored Procedures
 - Functions

Query 1

```

1 • USE Walmart;
2 • CREATE VIEW All_dom AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart LEFT JOIN services ON walmart.Walmart_id = services.ser...
3 • SELECT * FROM All_dom;
4
5

```

Result Grid

RegionName	City	DaysOnMarket
New York	New York	153
Fort Myers	Fort Myers	69
Laredo	Laredo	94
Los Angeles	Los Angeles	84
North Las Vegas	North Las Vegas	89
Chicago	Chicago	64
Philadelphia	Philadelphia	60
Phoenix	Phoenix	81
Las Vegas	Las Vegas	51
San Diego	San Diego	70
Dallas	Dallas	40
Fremont	Fremont	60
San Jose	San Jose	75
Fayetteville	Fayetteville	64
Arlington	Arlington	44
Jacksonville	Jacksonville	102
Irving	Irving	109
San Francisco	San Francisco	59
Austin	Austin	54

Object Info Session All_dom 9

Table: services

Action Output

Time	Action	Response	Duration / Fetch Time
19:00:32	SELECT * FROM Walmart.services LIMIT 0, 1000	1000 row(s) returned	0.0023 sec / 0.0071 s...
43 19:07:49	Apply changes to services	Changes applied	
45 19:08:59	CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON wa...	Error Code: 1054. Unknown column 'services.services...' in...	0.00094 sec
46 19:09:26	SELECT * FROM DOM_Phoenix LIMIT 0, 1000	3 row(s) returned	0.012 sec / 0.000011...
47 19:10:18	CREATE VIEW DOM_AS SELECT walmart.SizeRank, walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER...	0 row(s) affected	0.0024 sec
48 19:10:52	SELECT * FROM DOM LIMIT 0, 1000	0 row(s) returned	0.014 sec / 0.000012...
49 19:12:01	CREATE VIEW sales_services AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from ...	1000 row(s) returned	0.0017 sec
50 19:12:21	SELECT * FROM sales_services LIMIT 0, 1000	0 row(s) affected	0.0042 sec / 0.0060...
51 19:13:19	CREATE VIEW All_dom AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart LEFT JOIN services...	1000 row(s) returned	0.0027 sec
52 19:13:43	SELECT * FROM All_dom LIMIT 0, 1000	1000 row(s) returned	0.0071 sec / 0.0075...

Object Info Session All_dom 9

Table: services

Action Output

Time	Action	Response	Duration / Fetch Time
19:00:32	SELECT * FROM Walmart.services LIMIT 0, 1000	1000 row(s) returned	0.0023 sec / 0.0071 s...
43 19:07:49	Apply changes to services	Changes applied	
45 19:08:59	CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER JOIN services ON wa...	Error Code: 1054. Unknown column 'services.services...' in...	0.00094 sec
46 19:09:26	SELECT * FROM DOM_Phoenix LIMIT 0, 1000	3 row(s) returned	0.012 sec / 0.000011...
47 19:10:18	CREATE VIEW DOM_AS SELECT walmart.SizeRank, walmart.RegionName, walmart.City, services.DaysOnMarket from walmart INNER...	0 row(s) affected	0.0024 sec
48 19:10:52	SELECT * FROM DOM LIMIT 0, 1000	0 row(s) returned	0.014 sec / 0.000012...
49 19:12:01	CREATE VIEW sales_services AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from ...	1000 row(s) returned	0.0017 sec
50 19:12:21	SELECT * FROM sales_services LIMIT 0, 1000	0 row(s) affected	0.0042 sec / 0.0060...
51 19:13:19	CREATE VIEW All_dom AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart LEFT JOIN services...	1000 row(s) returned	0.0027 sec
52 19:13:43	SELECT * FROM All_dom LIMIT 0, 1000	1000 row(s) returned	0.0071 sec / 0.0075...

Query Completed

Local instance 3306

Administration Schemas Query 1 create-databases

SCHEMAS

Filter objects

- > sql_hr
- > sql_inventory
- > sql_invoicing
- > sql_store
- > sys
- Walmart**
 - Tables
 - > employment
 - services**
 - > walmart
 - Views
 - Stored Procedures
 - Functions

Result Grid Filter Rows: Search Export: Fetch rows:

```

1 • USE Walmart;
2 • CREATE VIEW Services_dom AS SELECT walmart.RegionName, walmart.City, walmart.State, services.DaysOnMarket FROM walmart LEFT JOIN services ON walmart.Walmar...
3 • SELECT * FROM Services_dom;
4
5

```

RegionName	City	State	DaysOnMarket
New York	New York	NY	153
Fort Myers	Fort Myers	FL	66
Laredo	Laredo	TX	84
Los Angeles	Los Angeles	CA	84
North Las Vegas	North Las Vegas	NV	89
Chicago	Chicago	IL	64
Philadelphia	Philadelphia	PA	60
Phoenix	Phoenix	AZ	81
Las Vegas	Las Vegas	NV	51
San Diego	San Diego	CA	70
Dallas	Dallas	TX	40
Fremont	Fremont	CA	90
San Jose	San Jose	CA	75
Fayetteville	Fayetteville	NC	64
Arlington	Arlington	VA	44
Jacksonville	Jacksonville	FL	102
Irving	Irving	TX	109
San Francisco	San Francisco	CA	59
Austin	Austin	TX	54
Orlando	Orlando	FL	22

Object Info Session Table: services

Action Output

Time	Action	Response	Duration / Fetch Time
40 19:09:09	CREATE VIEW DOM_Phoenix AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart INNER JOIN services ON walmart.Walmar...	0 rows(s) affected	0.000 sec / 0.00001...
46 19:09:26	SELECT * FROM DOM_Phoenix LIMIT 0, 1000	3 row(s) returned	0.012 sec / 0.00001...
47 19:10:18	CREATE VIEW DOM AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart INNER JOIN services ON walmart.Walmar...	0 row(s) affected	0.0024 sec
48 19:10:52	SELECT * FROM DOM LIMIT 0, 1000	0 row(s) returned	0.014 sec / 0.000012...
49 19:12:01	CREATE VIEW sales_services AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from ...	0 row(s) affected	0.001 sec
50 19:12:21	SELECT * FROM sales_services LIMIT 0, 1000	1000 row(s) returned	0.0042 sec / 0.0060...
51 19:13:19	CREATE VIEW All_dom AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart LEFT JOIN services...	0 row(s) affected	0.0027 sec
52 19:13:43	SELECT * FROM All_dom LIMIT 0, 1000	1000 row(s) returned	0.0071 sec / 0.00075...
53 19:14:28	CREATE VIEW Services_dom AS SELECT walmart.RegionName, walmart.City, walmart.State, services.DaysOnMarket FROM walmart...	0 row(s) affected	0.0046 sec
54 19:14:51	SELECT * FROM Services_dom LIMIT 0, 1000	1000 row(s) returned	0.053 sec / 0.00011 s...

Query Completed

Local instance 3306

Administration Schemas Query 1 create-databases

SCHEMAS

Filter objects

- > sql_hr
- > sql_inventory
- > sql_invoicing
- > sql_store
- > sys
- Walmart**
 - Tables
 - > employment
 - services**
 - > walmart
 - Views
 - Stored Procedures
 - Functions

Result Grid Filter Rows: Search Export: Fetch rows:

```

1 • USE Walmart;
2 • CREATE VIEW negative_equity AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from ... Walmart INNER JOIN services ON ...
3 • SELECT * FROM negative_equity;
4
5

```

RegionName	City	NegativeEquity	Delinquency
Boston	Boston	0.066470401567977	0.016734360365345

Object Info Session Table: services

Action Output

Time	Action	Response	Duration / Fetch Time
47 19:10:10	CREATE VIEW DOM AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart INNER JOIN services ON ...	0 row(s) affected	0.0024 sec
48 19:10:52	SELECT * FROM DOM LIMIT 0, 1000	0 row(s) returned	0.014 sec / 0.000012...
49 19:12:01	CREATE VIEW sales_services AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from ...	0 row(s) affected	0.0017 sec
50 19:12:21	SELECT * FROM sales_services LIMIT 0, 1000	1000 row(s) returned	0.0042 sec / 0.0060...
51 19:13:19	CREATE VIEW All_dom AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart LEFT JOIN services...	0 row(s) affected	0.0027 sec
52 19:13:43	SELECT * FROM All_dom LIMIT 0, 1000	1000 row(s) returned	0.0071 sec / 0.00075...
53 19:14:28	CREATE VIEW negative_equity AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from ...	0 row(s) affected	0.0046 sec
54 19:14:51	SELECT * FROM negative_equity LIMIT 0, 1000	1000 row(s) returned	0.053 sec / 0.00011 s...
55 19:15:42	CREATE VIEW Services_dom AS SELECT walmart.RegionName, walmart.City, walmart.State, services.DaysOnMarket FROM walmart...	1 row(s) affected	0.0033 sec
56 19:16:12	SELECT * FROM Services_dom LIMIT 0, 1000	1 row(s) returned	0.0087 sec / 0.00001...

Query Completed

Local instance 3306

Administration Schemas Query 1 create-databases

SCHEMAS

Filter objects

- > sql_hr
- > sql_inventory
- > sql_invoicing
- > sql_store
- > sys
- Walmart**
 - Tables
 - > employment
 - > services
 - > walmart
 - Views
 - Stored Procedures
 - Functions

Object Info Session ZHVI 12

Table: **services**

Action Output

Time	Action	Response	Duration / Fetch Time
49 19:14:19	CREATE VIEW SHVI_services AS SELECT walmart.RegionName, walmart.City, services.ZHVI FROM walmart INNER JOIN services ON walmart.Walmart_id = services.services_id	0 rows(s) affected	0.0001 sec
50 19:12:21	SELECT * FROM sales_services LIMIT 0, 1000	1000 row(s) returned	0.0042 sec / 0.0006...
51 19:13:19	CREATE VIEW All_dom AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart LEFT JOIN services...	0 row(s) affected	0.0027 sec
52 19:13:43	SELECT * FROM All_dom LIMIT 0, 1000	1000 row(s) returned	0.0071 sec / 0.00075...
53 19:14:28	CREATE VIEW Services_dom AS SELECT walmart.RegionName, walmart.City, walmart.State, services.DaysOnMarket FROM walmart...	0 row(s) affected	0.0046 sec
54 19:14:51	SELECT * FROM Services_dom LIMIT 0, 1000	1000 row(s) returned	0.053 sec / 0.0001 s...
55 19:15:42	CREATE VIEW negative_equity AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from...	0 row(s) affected	0.0033 sec
56 19:16:12	SELECT * FROM negative_equity LIMIT 0, 1000	1 row(s) returned	0.0087 sec / 0.00001...
57 19:16:55	CREATE VIEW ZHVI AS SELECT walmart.SellForGain, walmart.RegionName, walmart.City, services.ZHVI from walmart INNER JOIN services...	0 row(s) affected	0.0022 sec
58 19:18:01	SELECT * FROM ZHVI LIMIT 0, 1000	1 row(s) returned	0.0095 sec / 0.00001...

Query Completed

Local instance 3306

Administration Schemas Query 1 create-databases

SCHEMAS

Filter objects

- > sql_hr
- > sql_inventory
- > sql_invoicing
- > sql_store
- > sys
- Walmart**
 - Tables
 - > employment
 - > services
 - > walmart
 - Views
 - Stored Procedures
 - Functions

Object Info Session DOM_states 13

Table: **services**

Action Output

Time	Action	Response	Duration / Fetch Time
01 19:13:19	CREATE VIEW All_dom AS SELECT walmart.RegionName, walmart.City, services.DaysOnMarket FROM walmart LEFT JOIN services...	0 rows(s) affected	0.0027 sec
52 19:13:43	SELECT * FROM All_dom LIMIT 0, 1000	1000 row(s) returned	0.0071 sec / 0.00075...
53 19:14:28	CREATE VIEW Services_dom AS SELECT walmart.RegionName, walmart.City, walmart.State, services.DaysOnMarket FROM walmart...	0 row(s) affected	0.0046 sec
54 19:14:51	SELECT * FROM Services_dom LIMIT 0, 1000	1000 row(s) returned	0.053 sec / 0.0001 s...
55 19:15:42	CREATE VIEW negative_equity AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from...	0 row(s) affected	0.0033 sec
56 19:16:12	SELECT * FROM negative_equity LIMIT 0, 1000	1 row(s) returned	0.0087 sec / 0.00001...
57 19:16:55	CREATE VIEW DOM_states AS SELECT walmart.State, SUM(services.DaysOnMarket) from walmart INNER JOIN services ON walmart...	0 row(s) affected	0.0046 sec
58 19:18:01	SELECT * FROM DOM_states LIMIT 0, 1000	48 row(s) returned	0.025 sec / 0.00001...

Query Completed

Local Instance 3306

Administration Schemas Query 1 create-databases

SCHEMAS

Filter objects

- > sql_hr
- > sql_inventory
- > sql_invoicing
- > sql_store
- > sys
- Walmart
 - Tables
 - > employment
 - > services
 - > walmart
 - Views
 - Stored Procedures
 - Functions

100% 23:3 | Result Grid Filter Rows: Search Export:

```

USE Walmart;
CREATE VIEW min_MHI AS SELECT walmart.MarketHealthIndex, walmart.RegionName, walmart.City, services.MoM FROM walmart INNER JOIN services ON walmart.WalmartID = services.ID;
SELECT * FROM min_MHI;

```

MarketHealthIndex	RegionName	City	MoM
7.5552108048914	Los Angeles	Los Angeles	1.00791936645068
10	Wahneta	Wahneta	1.5974408945687

Object Info Session min_MHI 15 | Read Only

Table: services

Action Output

Time	Action	Response	Duration / Fetch Time
50 19:10:42	CREATE VIEW negative_equity AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency, sum(services.DaysOnMarket) FROM walmart INNER JOIN services ON walmart.WalmartID = services.ID;	0 rows(s) affected	0.0003 sec
56 19:16:12	SELECT * FROM negative_equity LIMIT 0, 1000	1 row(s) returned	0.0087 sec / 0.00001...
57 19:16:55	CREATE VIEW ZHVI AS SELECT walmart.SellForGain, walmart.RegionName, walmart.City, services.ZHVI FROM walmart INNER JOIN services ON walmart.WalmartID = services.ID;	0 row(s) affected	0.0022 sec
58 19:18:01	SELECT * FROM ZHVI LIMIT 0, 1000	1 row(s) returned	0.0095 sec / 0.00001...
59 19:18:56	CREATE VIEW DOM_states AS SELECT walmart.State, SUM(services.DaysOnMarket) from walmart INNER JOIN services ON walmart.WalmartID = services.ID;	0 row(s) affected	0.0046 sec
60 19:19:16	SELECT * FROM DOM_states LIMIT 0, 1000	48 row(s) returned	0.025 sec / 0.00001...
61 19:20:12	CREATE VIEW weekly_sales AS SELECT Store, Weekly_Sales, Date from employment where Date = 05-02-2010 AND Weekly_Sales >= (SELECT MAX(Weekly_Sales) from employment WHERE Date = 05-02-2010)	0 row(s) affected	0.0048 sec
62 19:20:33	SELECT * FROM weekly_sales LIMIT 0, 1000	0 row(s) returned	0.014 sec / 0.000010...
63 19:21:21	CREATE VIEW min_MHI AS SELECT walmart.MarketHealthIndex, walmart.RegionName, walmart.City, services.MoM from walmart INNER JOIN services ON walmart.WalmartID = services.ID;	0 row(s) affected	0.0051 sec
64 19:21:41	SELECT * FROM min_MHI LIMIT 0, 1000	2 row(s) returned	0.021 sec / 0.000010...

Query Completed

Local Instance 3306

Administration Schemas Query 1 create-databases

SCHEMAS

Filter objects

- > sql_hr
- > sql_inventory
- > sql_invoicing
- > sql_store
- > sys
- Walmart
 - Tables
 - > employment
 - > services
 - > walmart
 - Views
 - Stored Procedures
 - Functions

100% 28:3 | Result Grid Filter Rows: Search Export:

```

USE Walmart;
CREATE VIEW weekly_sales AS SELECT Store, Weekly_Sales, Date from employment where Date = 05-02-2010 AND Weekly_Sales=(SELECT MAX(Weekly_Sales) from employment WHERE Date = 05-02-2010);
SELECT * FROM weekly_sales;

```

Store	Weekly_Sales	Date
-------	--------------	------

Object Info Session weekly_sales 14 | Read Only

Table: services

Action Output

Time	Action	Response	Duration / Fetch Time
53 19:14:20	CREATE VIEW services_join AS SELECT walmart.RegionName, walmart.City, services.Dom LIMIT 0, 1000	1000 rows(s) returned	0.0040 sec
54 19:14:51	SELECT * FROM Services_dom LIMIT 0, 1000	0 row(s) affected	0.0033 sec
55 19:15:42	CREATE VIEW negative_equity AS SELECT walmart.RegionName, walmart.City, services.NegativeEquity, services.Delinquency from ...	0 row(s) affected	0.0087 sec / 0.00001...
56 19:16:12	SELECT * FROM negative_equity LIMIT 0, 1000	1 row(s) returned	0.0022 sec
57 19:16:55	CREATE VIEW ZHVI AS SELECT walmart.SellForGain, walmart.RegionName, walmart.City, services.ZHVI FROM walmart INNER JOIN services ON walmart.WalmartID = services.ID;	0 row(s) affected	0.0046 sec
58 19:18:01	SELECT * FROM ZHVI LIMIT 0, 1000	1 row(s) returned	0.0095 sec / 0.00001...
59 19:18:56	CREATE VIEW DOM_states AS SELECT walmart.State, SUM(services.DaysOnMarket) from walmart INNER JOIN services ON walmart.WalmartID = services.ID;	48 row(s) returned	0.025 sec / 0.00001...
60 19:19:16	SELECT * FROM DOM_states LIMIT 0, 1000	0 row(s) affected	0.0048 sec
61 19:20:12	CREATE VIEW weekly_sales AS SELECT Store, Weekly_Sales, Date from employment where Date = 05-02-2010 AND Weekly_Sales >= (SELECT MAX(Weekly_Sales) from employment WHERE Date = 05-02-2010)	0 row(s) affected	0.014 sec / 0.000010...
62 19:20:33	SELECT * FROM weekly_sales LIMIT 0, 1000	0 row(s) returned	0.014 sec / 0.000010...

Query Completed

