

Team Members :
 Abhi Mehta 221080001
 Aayush Khiraiyya 221080037
 Branch: SY IT 2022-26
 COURSE INSTRUCTOR : V.B. Nikam Sir

Assignment-02

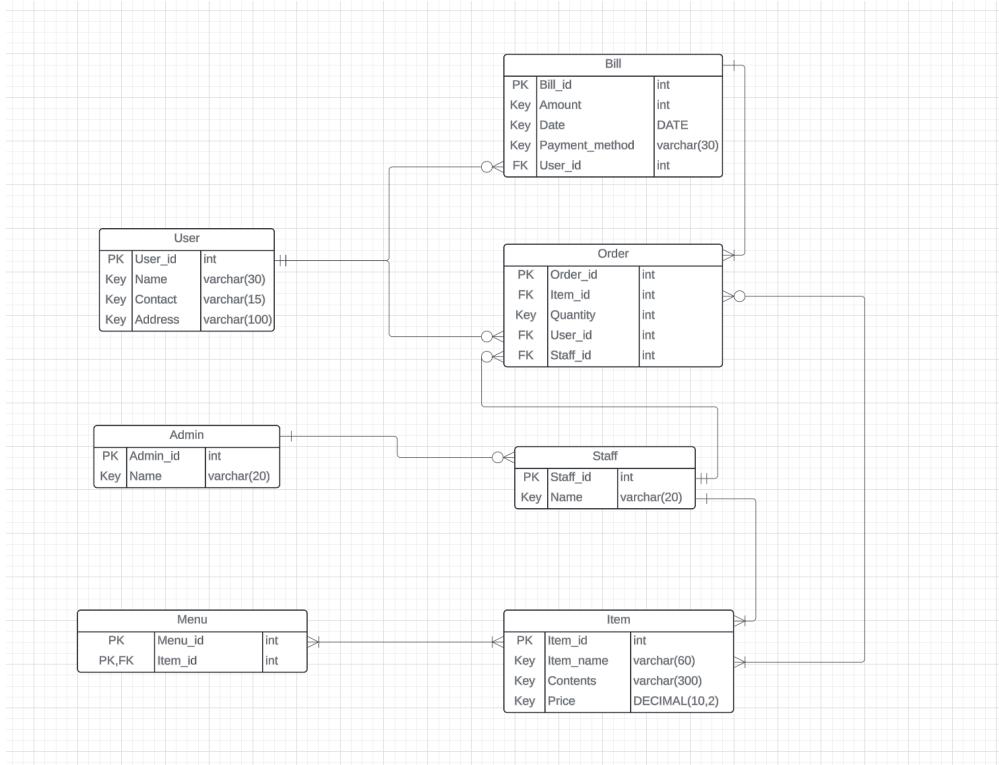
- Aim-1.** Normalize all the schema to 3NF/BCNF.
 2. You should prepare/collect data required for your problem statement, at least 50 records for each table.
 3. Write a list of questions (at least 100) which can be asked to your solution system through SQL statements, and the answer of the question is the Result of the SQL query written for that question.

Changes to previous ER diagram:

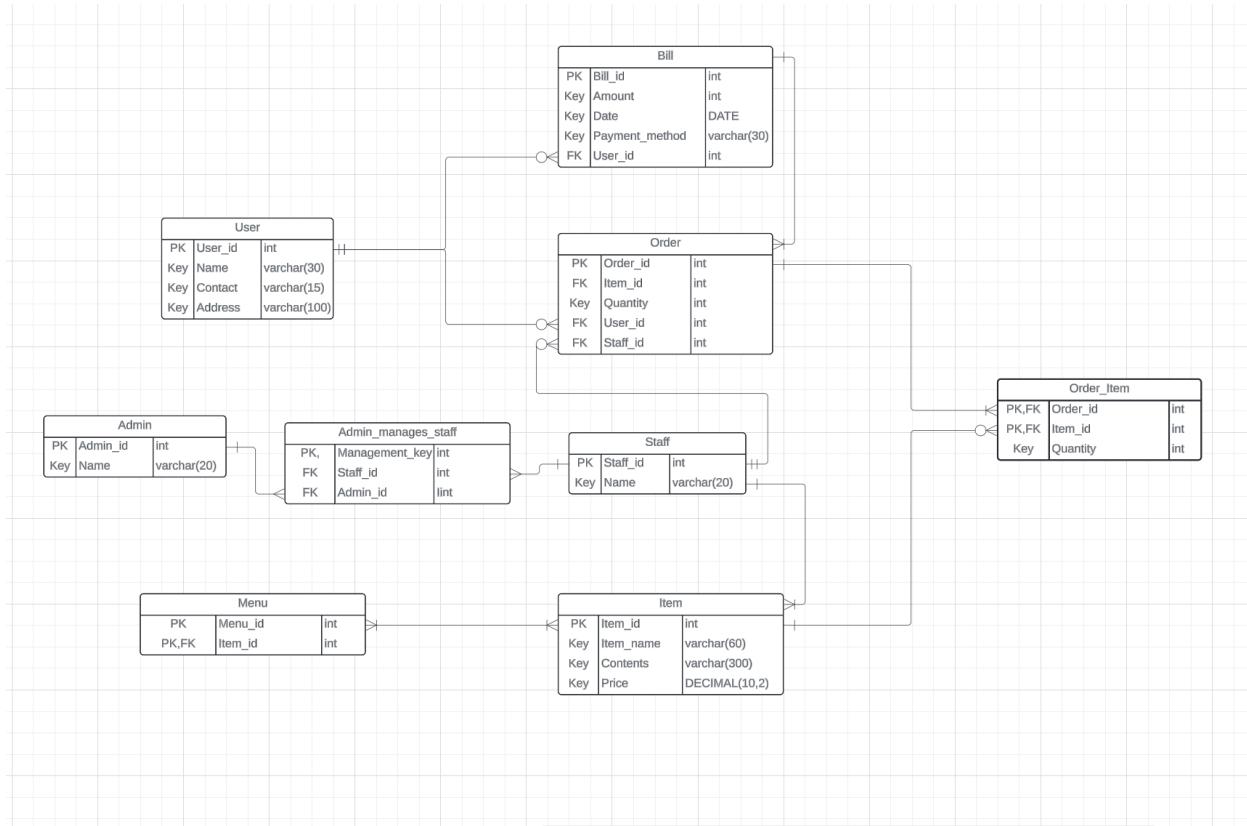
Based on our requirements we have made some necessary changes in the tables of our database management system:

1. Addition of Staff_id to Order table as foreign key to relate staff with orders.
2. Admin has a role of managing the DBMS and staff rather than generating bills only.
3. Many to many relation between Admin and Staff

Logical Schema:



Normalization of Schema to 3NF/BCNF:



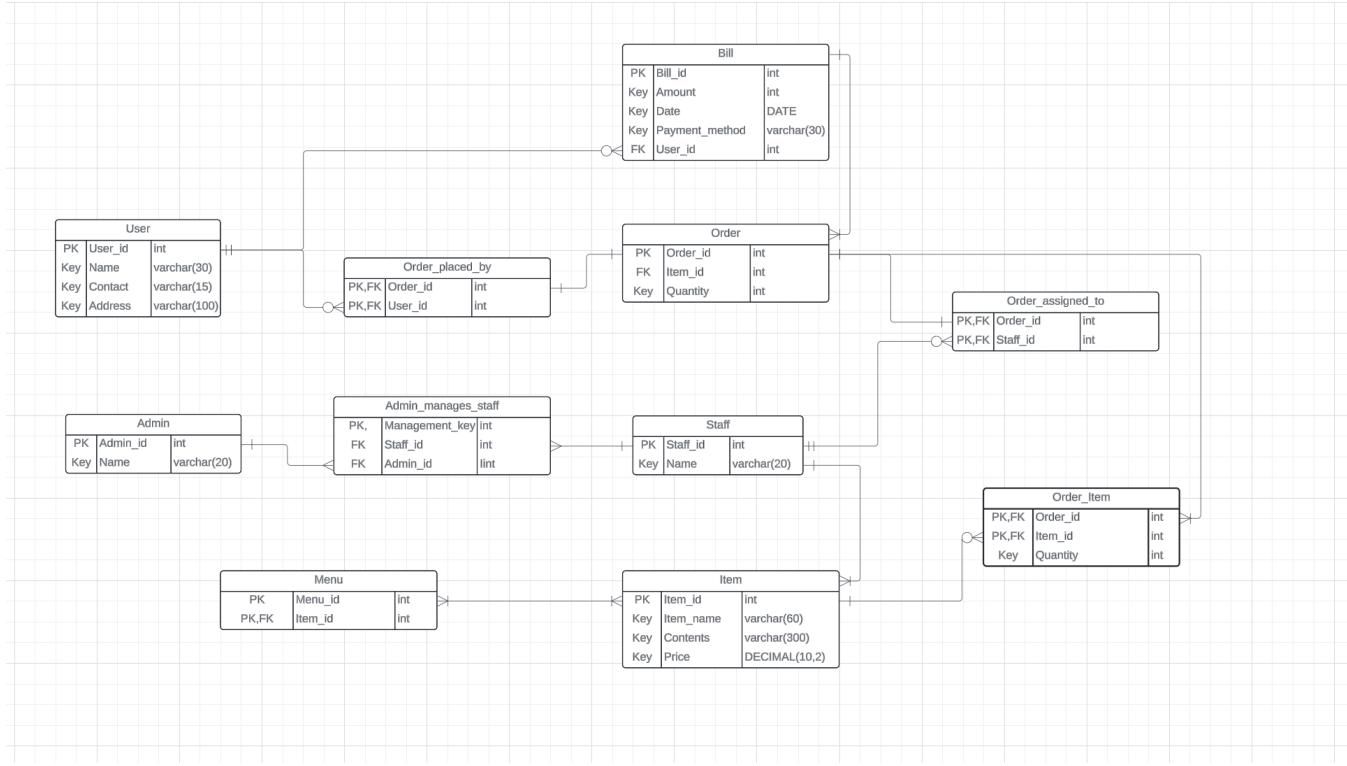
The 3rd Normal Form (3NF) and Boyce-Codd Normal Form (BCNF) analysis on each table in the provided schema:

1. Menu table:
 - This table is already in BCNF because the only determinant is the composite primary key (Menu_id, Item_id).
2. Item table:
 - This table is in 3NF and BCNF because the primary key (Item_id) is the only determinant.
 - The Key (Item_name, Contents, Price) is a candidate key, which satisfies BCNF.
3. User table:
 - This table is in 3NF and BCNF because the primary key (User_id) is the only determinant.
 - The Key (Name, Contact, Address) is a candidate key, which satisfies BCNF.

4. Staff table:
 - This table is in 3NF and BCNF because the primary key (Staff_id) is the only determinant.
 - The Key (Name) is not a candidate key, but there is no non-key attribute determined by Name, so it still satisfies BCNF.
5. Order table:
 - This table is not in 3NF and BCNF because the primary key (Order_id) is not the only determinant.
 - Order & Order_Item: Both tables have potential 3NF violations. Order_id alone might not determine all attributes in Order (similarly for Order_id and Quantity in Order_Item).
6. Order_Item table:
 - This table is already in BCNF because the only determinant is the composite primary key (Order_id, Item_id).
 - The Key (Quantity) is not a candidate key, but there is no non-key attribute determined by Quantity, so it still satisfies BCNF.
7. Admin table:
 - This table is in 3NF and BCNF because the primary key (Admin_id) is the only determinant.
 - The Key (Name) is not a candidate key, but there is no non-key attribute determined by Name, so it still satisfies BCNF.
8. Bill table:
 - This table is in 3NF and BCNF because the primary key (Bill_id) is the only determinant.
 - The Key (Amount, Date, Payment_method) is not a candidate key, but there is no non-key attribute determined by this set of attributes, so it still satisfies BCNF.
9. Admin_manages_staff table:
 - This table is already in BCNF because the only determinant is the primary key (Management_key).

Based on the analysis, all tables in the provided schema are not satisfying the 3rd Normal Form (3NF) and the Boyce-Codd Normal Form (BCNF).

Final Schema:



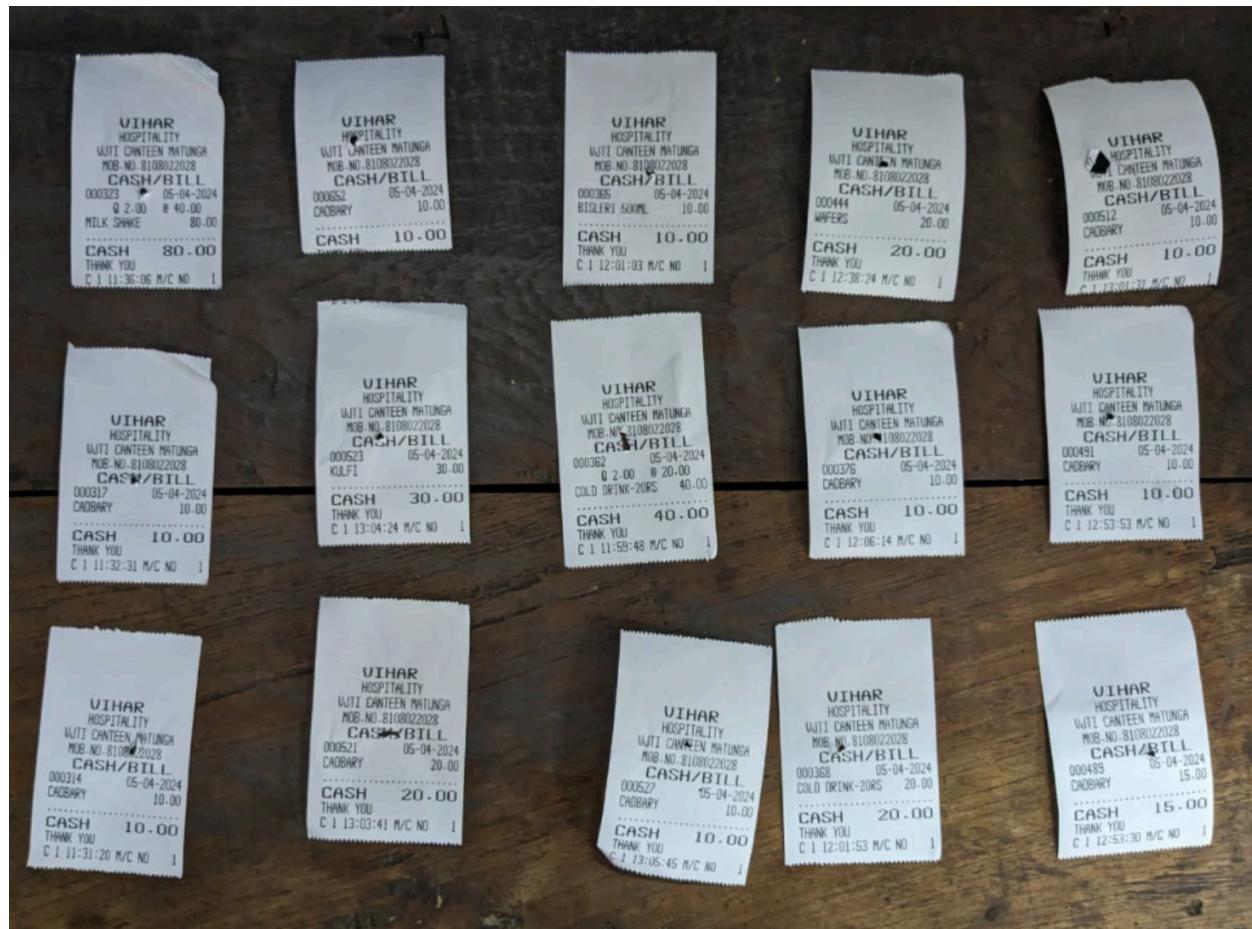
- Normalization:
 - 3NF: All tables are now in 3NF. The primary keys uniquely identify each row, and all attributes are fully dependent on them.
 - BCNF: The schema also adheres to BCNF due to the presence of surrogate keys in Management_key (Admin_manages_staff) and the removal of the composite primary key in Order.
 -
 - Changes:
 - Order: The primary key in Order is now simply Order_id. This avoids the potential 3NF violation from the previous version where including User_id and Staff_id might have introduced redundancy.
 - Order_placed_by & Order_assigned_to: These new tables handle the many-to-many relationships between Order, User, and Staff. Each table has a composite primary key consisting of Order_id and the respective foreign key (User_id or Staff_id). This approach ensures efficient linking and maintains data integrity.

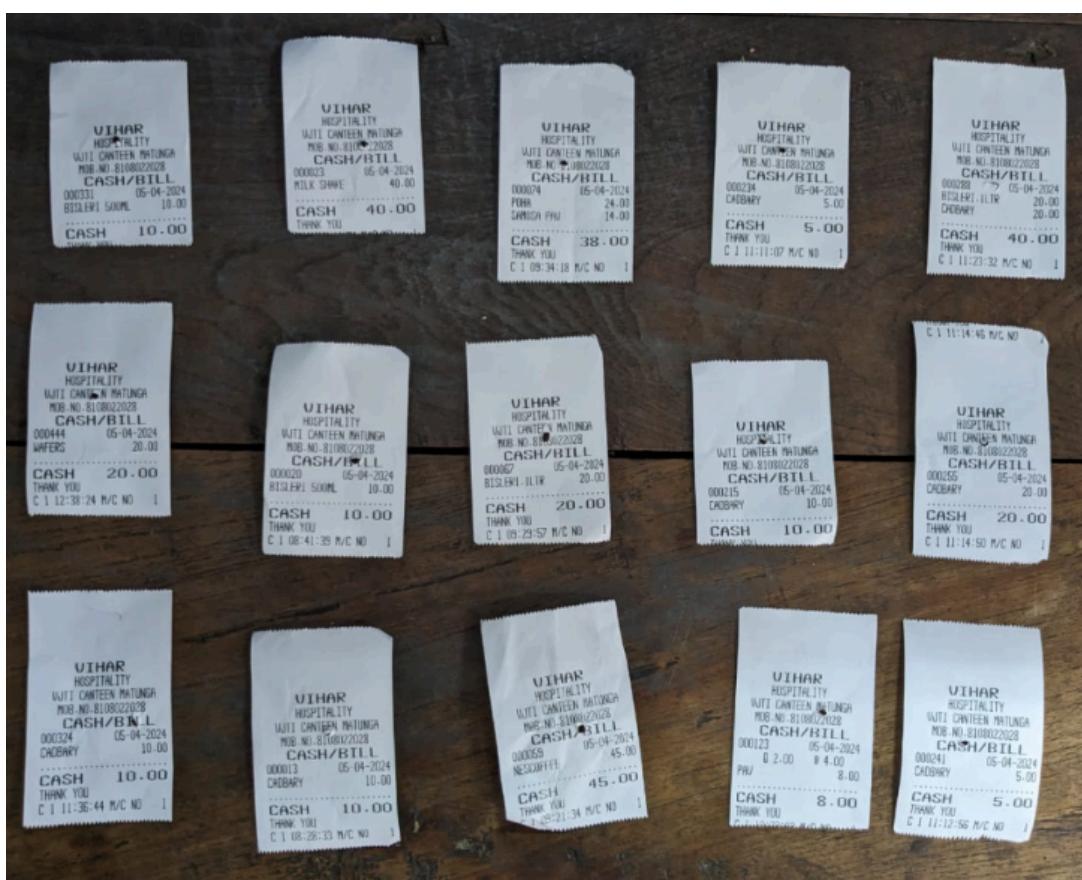
Data Collected:

Raw data arranged in Excel file for each table:

Order ID	Item ID	Item Name	Quantity	Unit Price	Total Price	Date	Customer ID	Customer Name	Address	Phone Number	Order Status	Notes
2210	3	1 Aayushi	7719865551	Aysha Beder, Cobbs, Mumbai, Maharashtra 400091, India	2200	5	S-Air Cash	Vada Pav	Vadeavish	14	241	1
2210	2	2 Aditi	7719865521	, 136, Dr E Moses Rd, Gunda Nagar, Upper Work, Ward, Mumbai, Maharashtra 400018, India	2201	8	S-Air Cash	Samosa Pav	Samosa pav	14	123	2
2210	3	3 Aishwarya	7719865541	Jata Tava Rd, Uda Tarma Housing Colony, Sector 1, Tava, Jata, Mumbai, Maharashtra 400049, India	2202	45	S-Air Cash	Vada Plate	Vadaplate	22	59	1
2210	4	4 Aishwarya	7719865544	Shivaji Nagar, Mumbai, Maharashtra 400099, India	2203	10	S-Air Cash	Poha Plate	Poha plate	24	33	1
2210	5	5 Aishwarya	7719865555	Bahay Sahay Marg, Andheri East, Andheri, Karla Rd, Andheri East, Mumbai, Maharashtra 400072, India	2204	10	S-Air Cash	Poha	Poha is a tr	25	74	1
2210	6	6 Aastha	7719865556	Sakshi Jantek, Andheri, Andheri Rd, Andheri East, Mumbai, Maharashtra 400072, India	2205	20	S-Air Cash	Idli-Sambar	Idli sambar	24	255	1
2210	7	7 Aashna	7719865557	Shivaji Nagar, Mumbai, Maharashtra 400099, India	2206	10	S-Air Cash	Uppma	Uppma	25	215	1
2210	8	8 Aishwarya	7719865558	Opposite Mint International Tower, Andheri - Karla Rd, Andheri East, Mumbai, Maharashtra 400099, India	2207	5	S-Air Cash	Uppma Poha	Uppma pohi	28	67	1
2210	9	9 Aachal	7719865559	Andheri - Karla Rd, JB Nagar, Andheri East, Mumbai, Maharashtra 400099, India	2208	10	S-Air Cash	Medu Vada	Medu vada	26	20	1
2210	10	10 Arani	7719865560	Mint International Airport, TA Project Rd, Navi Mumbai, Vile Parle East, Vile Parle, Andheri, Maharashtra 400099, India	2209	10	S-Air Cash	Instant Poha	Instant poha	29	444	1
2210	11	11 Disha	7620489261	Aysha Beder, Cobbs, Mumbai, Maharashtra 400091, India	2210	40	S-Air Cash	Instant Poha	Instant poha	31	298	2
2210	12	12 Geetanjali	7620489262	opp. Domestic Airport, Navi Mumbai, Vile Parle East, Vile Parle, Mumbai, Maharashtra 400099, India	2211	5	S-Air Cash	Vada Usa	Vadeasas	29	234	1
2210	13	13 Maitri	7620489263	Sakshi Vilas Rd, Miraoad Nagar, Miraoad, Panvel, Powai, Mumbai, Maharashtra 400087, India	2212	38	S-Air Cash	Ragda Samosa	Ragda Sam	35	74	2
2210	14	14 Maitri	7620489264	Opposite Kharadi Chawl, Sector 10, Kharadi, Andheri East, Mumbai, Maharashtra 400099, India	2213	40	S-Air Cash	Ragda Samosa	Ragda Sam	39	271	1
2210	15	15 Mrunmayi	7620489265	462, Sengupta Baipat Marg, Lower Patel, Mumbai, Maharashtra 400091, India	2214	10	S-Air Cash	Veg Sandwich	Veg sandwich	33	331	1
2210	16	16 Nishit	7620489266	Plot No. 31, MIDC Central Rd, near Alankar Convent, Chakala Industrial Area (MIDC), Andheri East, Mumbai, Maharashtra 400093, India	2215	10	S-Air Cash	Veg Cheese Sandwich	Veg cheese sandwich	50	710	1
2210	17	17 Pavan	7620489267	Opposite Oberoi Garden City, International Business Park, Fabholon, Goregaon, Mumbai, Maharashtra 400099, India	2216	10	S-Air Cash	Mumbai Sandwich	Mumbai sandwich	33	723	1
2210	18	18 Prachi	7620489268	Opposite Oberoi Garden City, International Business Park, Fabholon, Goregaon, Mumbai, Maharashtra 400099, India	2217	15	S-Air Cash	Mumbai Cheese Sandwich	Mumbai cheese sandwich	42	737	1
2210	19	19 Raj	7620489269	X, 22, MIDC Central Rd, Hinman Nagar, Andheri East, Mumbai, Maharashtra 400093, India	2218	10	S-Air Cash	Cheese Sandwich	Cheese sandwich	42	713	1
2210	20	20 Rohith	7620489270	Conifer School, Kandivali Chambers, Off Saler Rd, near Boway, AAI Colony, JB Nagar, Andheri East, Mumbai, Maharashtra 400099, India	2219	30	S-Air Cash	Veg Fried Rice	Veg fried rice	50	700	3
2210	21	21 Sushma	7620489271	Opposite Oberoi Garden City, International Business Park, Fabholon, Goregaon, Mumbai, Andheri West, Mumbai, Maharashtra 400095, India	2220	20	S-Air Cash	Veg Fried Rice	Veg fried rice	55	663	1
2210	22	22 Sejal	7620489272	333, Dr Antedhak Rd, Pal Pather, Khar West, Mumbai, Maharashtra 400052, India	2221	20	S-Air Cash	Veg Fafda Noodles	Vegitable I	50	661	1
2210	23	23 Sarvesh	7620489273	SV Rd, sent to Khar Majid, Khar, Dadasaheb, Mumbai, Maharashtra 400057, India	2222	25	S-Air Cash	Veg Schezwan Noodles	Vegetable I	55	844	1
2210	24	24 Shreya	7620489274	SV Rd, sent to Khar Majid, Khar, Dadasaheb, Mumbai, Maharashtra 400057, India	2223	10	S-Air Cash	Vada Dosa	Vada dosa	22	804	1
2210	25	25 Shweta	7620489275	Saga Future Garden City, Opp. Kharadi Chawl, Sector 10, Kharadi, Andheri East, Mumbai, Maharashtra 400099, India	2224	10	S-Air Cash	Masala Dosa	Masala dosa	33	800	1
2210	26	26 Shubham	7620489276	505, Mohamed I, Royal Palm Estate, Avery Colony, Goregaon(E), Mumbai, Maharashtra 400095, India	2225	20	S-Air Cash	Butter Sada Dosa	Batura dosa	29	731	1
2210	27	27 Sumrit	7620489277	India House No. 2, Near India Bank Pedder Road, Kemps Corner Flyover, Cambulla Hill, Mumbai, Maharashtra 400026, India	2226	20	S-Air Cash	Butter Masala Dosa	Butter masala dosa	39	703	1
2210	28	28 Vaibhavi	7620489278	Opp. Kharadi Chawl, Sector 10, Kharadi, Andheri East, Mumbai, Maharashtra 400099, India	2227	40	S-Air Cash	Masala Dosa	Masala dosa	29	674	2
2210	29	29 Vansh	7620489279	Shop No. 7, 11th Rd, Kond Kanji, Bandra West, Mumbai, Maharashtra 400095, India	2228	28	S-Air Cash	Myore Sada Dosa	Myore dosa	35	670	1
2210	30	30 Yash	7620489280	57, 21st Rd, Bandra West, Mumbai, Maharashtra 400095, India	2229	60	S-Air Cash	Schezwan Sada Dosa	Schezwan dosa	31	909	3
2210	31	31 Yashika	7620489281	Opposite Oberoi Garden City, International Business Park, Fabholon, Goregaon, Mumbai, Maharashtra 400099, India	2230	20	S-Air Cash	Schezwan Masala Dosa	Schezwan masala dosa	25	871	1
2210	32	32 Zeevishi	7620489282	201, Monsoon, 10th Floor, New Bhoomi, Sector 10, Kharadi, Andheri East, Mumbai, Maharashtra 400099, India	2231	20	S-Air Cash	Uttapam	Uttapam	24	939	2
2210	33	33 Rushi	7620489283	Manor Mania, Shop No. 115, 1st Main Street, Andheri East, Mumbai, Maharashtra 400091, India	2232	20	S-Air Cash	Onion Uttapam	Uttapam is	33	878	1
2210	34	34 Akash	7620489284	14A, SV Rd, sent to Khar Majid, Khar, Dadasaheb, Mumbai, Maharashtra 400057, India	2233	20	S-Air Cash	Onion Uttapam	Uttapam is	33	704	1
2210	35	35 Akash	7620489285	Opp. Kharadi Chawl, Sector 10, Kharadi, Andheri East, Mumbai, Maharashtra 400099, India	2234	10	S-Air Cash	Chivda Dosa	Chivda dosa	35	694	1
2210	36	36 Soham	7620489286	Shop No. 1, Pavon Bharat Chay Da Saay Chai, Dr Antedhak Marg, Andheri, Kadechvadi, Mumbai, Maharashtra 400031, India	2235	90	S-Air Cash	Lunch is r	Lunch is r	64	664	3
2210	37	37 Chatrali	7620489287	Shop No. 2, Jai Ambre, par, K.N. Murg, No. 8 Bank of India, Chembur, Mumbai, Maharashtra 400071, India	2236	80	S-Air Cash	Tea	Tea is r	11	323	3
2210	38	38 Chatrali	7620489288	Shop No. 2, Jai Ambre, par, K.N. Murg, No. 8 Bank of India, Chembur, Mumbai, Maharashtra 400071, India	2237	10	S-Air Cash	Coffee	Coffee is r	15	651	1
2210	39	39 Lubhi	7620489289	Double Pakhi Shop 3, Dr Antedhak Rd, Khar West, Mumbai, Maharashtra 400095, India	2238	30	S-Air Cash	Chassi	Chassi also	13	523	1
2210	40	40 Yuvraj	7620489210	Russon Co, Arasala Building, 3rd Floor, HDP Cine, Opp. Cobbs, Mumbai, Maharashtra 400095, India	2239	40	S-Air Cash	Lassi	Lassi is a tr	28	362	3

Raw Data:(Receipts)







Billing Info:

5/4/2024	
5/4/2024	
11	43
12	59
13	26
14	24
15	11
16	11
17	11
18	11
19	11
20	11
21	11
22	11
23	11
24	11
25	11
26	11
27	11
28	11
29	11
30	11
31	11
32	11
33	11
34	11
35	11
36	11
37	11
38	11
39	11
40	11
41	11
42	11
43	11
44	11
45	11
46	11
47	11
48	11
49	11
50	11
51	11
52	11
53	11
54	11
55	11
56	11
57	11
58	11
59	11
60	11
61	11
62	11
63	11
64	11
65	11
66	11
67	11
68	11
69	11
70	11
71	11
72	11
73	11
74	11
75	11
76	11
77	11
78	11
79	11
80	11
81	11
82	11
83	11
84	11
85	11
86	11
87	11
88	11
89	11
90	11
91	11
92	11
93	11
94	11
95	11
96	11
97	11
98	11
99	11
100	11
101	11
102	11
103	11
104	11
105	11
106	11
107	11
108	11
109	11
110	11
111	11
112	11
113	11
114	11
115	11
116	11
117	11
118	11
119	11
120	11
121	11
122	11
123	11
124	11
125	11
126	11
127	11
128	11
129	11
130	11
131	11
132	11
133	11
134	11
135	11
136	11
137	11
138	11
139	11
140	11
141	11
142	11
143	11
144	11
145	11
146	11
147	11
148	11
149	11
150	11
151	11
152	11
153	11
154	11
155	11
156	11
157	11
158	11
159	11
160	11
161	11
162	11
163	11
164	11
165	11
166	11
167	11
168	11
169	11
170	11
171	11
172	11
173	11
174	11
175	11
176	11
177	11
178	11
179	11
180	11
181	11
182	11
183	11
184	11
185	11
186	11
187	11
188	11
189	11
190	11
191	11
192	11
193	11
194	11
195	11
196	11
197	11
198	11
199	11
200	11
201	11
202	11
203	11
204	11
205	11
206	11
207	11
208	11
209	11
210	11
211	11
212	11
213	11
214	11
215	11
216	11
217	11
218	11
219	11
220	11
221	11
222	11
223	11
224	11
225	11
226	11
227	11
228	11
229	11
230	11
231	11
232	11
233	11
234	11
235	11
236	11
237	11
238	11
239	11
240	11
241	11
242	11
243	11
244	11
245	11
246	11
247	11
248	11
249	11
250	11
251	11
252	11
253	11
254	11
255	11
256	11
257	11
258	11
259	11
260	11
261	11
262	11
263	11
264	11
265	11
266	11
267	11
268	11
269	11
270	11
271	11
272	11
273	11
274	11
275	11
276	11
277	11
278	11
279	11
280	11
281	11
282	11
283	11
284	11
285	11
286	11
287	11
288	11
289	11
290	11
291	11
292	11
293	11
294	11
295	11
296	11
297	11
298	11
299	11
300	11
301	11
302	11
303	11
304	11
305	11
306	11
307	11
308	11
309	11
310	11
311	11
312	11
313	11
314	11
315	11
316	11
317	11
318	11
319	11
320	11
321	11
322	11
323	11
324	11
325	11
326	11
327	11
328	11
329	11
330	11
331	11
332	11
333	11
334	11
335	11
336	11
337	11
338	11
339	11
340	11
341	11
342	11
343	11
344	11
345	11
346	11
347	11
348	11
349	11
350	11
351	11
352	11
353	11
354	11
355	11
356	11
357	11
358	11
359	11
360	11
361	11
362	11
363	11
364	11
365	11
366	11
367	11
368	11
369	11
370	11
371	11
372	11
373	11
374	11
375	11
376	11
377	11
378	11
379	11
380	11
381	11
382	11
383	11
384	11
385	11
386	11
387	11
388	11
389	11
390	11
391	11
392	11
393	11
394	11
395	11
396	11
397	11
398	11
399	11
400	11
401	11
402	11
403	11
404	11
405	11
406	11
407	11
408	11
409	11
410	11
411	11
412	11
413	11
414	11
415	11
416	11
417	11
418	11
419	11
420	11
421	11
422	11
423	11
424	11
425	11
426	11
427	11
428	11
429	11
430	11
431	11
432	11
433	11
434	11
435	11
436	11
437	11
438	11
439	11
440	11
441	11
442	11
443	11
444	11
445	11
446	11
447	11
448	11
449	11
450	11
451	11
452	11
453	11
454	11
455	11
456	11
457	11
458	11
459	11
460	11
461	11
462	11
463	11
464	11
465	11
466	11
467	11
468	11
469	11
470	11
471	11
472	11
473	11
474	11
475	11
476	11
477	11
478	11
479	11
480	11
481	11
482	11
483	11
484	11
485	11
486	11
487	11
488	11
489	11
490	11
491	11
492	11
493	11
494	11
495	11
496	11
497	11
498	11
499	11
500	11

5/4/2024	
5/4/2024	
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68

Menu and Rates:

NEW RATES FROM 01 FEB - 2024	
VADA PAV - 14	SADA DOSA - 23
SAMOSA PAV - 14	MASALA DOSA - 33
VADA PLATE - 22	BUTTER SADA DOSA - 29
SAMOSA PLATE - 24	BUTTER MASALA DOSA - 39
POHA - 24	MISAL SADA DOSA - 29
IDLI SAMOSA - 24	MISAL MASALA DOSA - 35
UPMA - 25	SCHEZWAN SADA DOSA - 31
UPMA & POHA - 28	SCHEZWAN MASALA DOSA - 35
MEGU VADA - 26	SADA UTTAPAM - 24
BATATA VADA	ONION UTTAPAM - 33
SAMBER - 29	MASALA UTTAPAM - 33
MISAL PAV - 31	CHINESE DOSA - 53
VADA USAL - 29	LUNCH - 64
RAGADA SAMOSA - 35	
SAMOSA USAL - 35	
VEG SANDWICH - 33	
VEG CHEESE SW - 50	TEA - 11
MUMBAI SANDWICH - 33	COFFEE - 15
MUMBAI CHEESE SW - 42	CHASS - 13
CHEESE SANDWICH - 42	LAASS - 28
VEG FRIED RICE - 50	KOKAMSAR - 11
VEG SCHEZWAN RICE - 55	BOURN VITA - 22
VEG HAKKA NOODLES - 50	
VEG SCHEZWAN NOODLES - 55	

IDLI FRY - 25	
SEV PURI - 28	
VADA PAV - 14	
SAMOSA PAV - 14	
MISAL PAV - 31	
VADA USAL - 29	VEG SANDWICH - 33
BATATA VADA SAMOSA - 29	MUMBAI SW - 50
RAGADA SAMOSA - 35	VEG CHEESE SW - 50
SAMOSA USAL - 35	AGUMBAT
	CHEESE SW - 42

Queries :

1. Select all items from the Menu table.

Menu_id	Item_id
2210	1
2210	2
2210	3
2210	4
2210	5
2210	6
2210	7
2210	8
2210	9
2210	10
2210	11
2210	12
2210	13
2210	14
2210	15
2210	16
2210	17
2210	18
2210	19
2210	20
2210	21
2210	22
2210	23
2210	24
2210	25
2210	26
2210	27
2210	28
2210	29
2210	30
2210	31
2210	32
2210	33
2210	34
2210	35
2210	36
2210	37
2210	38
2210	39
2210	40

2.Select all orders from the Order table.

```
MariaDB [canteen_management]> Select * From `Order`;
```

Order_id	Item_id	Quantity
13	4	1
20	9	1
23	14	1
59	3	1
67	8	1
74	13	2
123	2	2
215	7	1
234	12	1
241	1	1
255	6	1
288	11	2
323	37	3
324	5	1
331	15	1
362	40	3
444	10	1
523	39	1
652	38	1
661	22	1
663	21	1
664	36	3
670	29	1
672	28	2
694	35	1
700	20	3
703	27	1
704	34	1
710	16	1
713	19	1
717	18	1
721	17	1
731	26	1
800	25	1
844	23	1
849	24	1
878	33	1
879	31	1
909	30	3
939	32	2

```
40 rows in set (0.001 sec)
```

3. Select all item names, contents, and prices from the Item table.

```
MariaDB [canteen_management]> SELECT Item_name, Contents, Price FROM Item;
+-----+-----+
| Item_name | Contents | Price |
+-----+-----+
| Batata Wada | Batatawada is a popular Indian street food consisting of spicy mashed potato filling coated in chickpea flour batter and deep-fried until crispy. | 29.00 |
| Butter Masala Dosa | Masala dosa is a South Indian crepe made from fermented rice and lentil batter, filled with a spiced potato filling and served with chutneys and sambar. | 39.00 |
| Butter Sada Dosa | Masala dosa is a South Indian crepe made from fermented rice and lentil batter, filled with a spiced potato filling and served with chutneys and sambar. | 29.00 |
| Chass | Chaas, also known as buttermilk, is a refreshing Indian drink made by diluting yogurt with water and flavored with spices like cumin, salt, and sometimes mint, offering a tangy and cooling sensation. | 13.00 |
| Cheese Sandwich | A veg sandwich typically consists of various vegetables such as lettuce, tomatoes, cucumbers, and bell peppers, layered between slices of bread, often accompanied by spreads like mayo or mustard. | 42.00 |
| Chinese Dosa | A Chinese dosa is a fusion dish that combines the traditional South Indian dosa with Chinese flavors, featuring fillings like stir-fried vegetables, sauces such as soy or chili sauce, and sometimes toppings like paneer or tofu. | 53.00 |
| Coffe | Coffee is a brewed beverage made from roasted coffee beans, enjoyed for its rich, aromatic flavor and stimulating effects. | 15.00 |
| Idli-Sambhar | Idli sambhar is a traditional South Indian dish consisting of steamed rice cakes (idli) served with a flavorful lentil-based stew (sambhar). | 24.00 |
| Lassi | Lassi is a traditional Indian yogurt-based drink, often flavored with fruits or spices, and served chilled. | 28.00 |
| Lunch | Lunch is a midday meal typically eaten between late morning and early afternoon, often consisting of a balanced combination of proteins, carbohydrates, and vegetables. | 64.00 |
| Masala Dosa | Masala dosa is a South Indian crepe made from fermented rice and lentil batter, filled with a spiced potato filling and served with chutneys and sambar. | 33.00 |
| Masala Uttappa | Uttapam is a savory South Indian pancake made from a fermented batter of rice and lentils, typically topped with onions, tomatoes, chilies, and sometimes other vegetables, cooked on a griddle. | 33.00 |
| Medu Wada | Medu vada is a South Indian fried snack made from a batter of urad dal (black gram) seasoned with spices and herbs, typically served with chutneys like coconut chutney and sambar. | 26.00 |
| Misal Pav | Misal pav is a spicy Maharashtrian street food dish consisting of sprouted lentils cooked in a fiery gravy, served with pav (bread rolls), topped with various garnishes like farsan (crunchy mix), onions, and coriander. | 31.00 |
| Mumbai Cheese Sandwich | A veg sandwich typically consists of various vegetables such as lettuce, tomatoes, cucumbers, and bell peppers, layered between slices of bread, often accompanied by spreads like mayo or mustard. | 42.00 |
| Mumbai Sandwich | A veg sandwich typically consists of various vegetables such as lettuce, tomatoes, cucumbers, and bell peppers, layered between slices of bread, often accompanied by spreads like mayo or mustard. | 31.00 |
```

Mumbai Sandwich	A veg sandwich typically consists of various vegetables such as lettuce, tomatoes, cucumbers, and bell peppers, layered between slices of bread, often accompanied by spreads like mayonnaise or mustard.
32.00	32.00
Mysore Masala Dosa	Masala dosa is a South Indian crepe made from fermented rice and lentil batter, filled with a spiced potato filling and served with chutneys and sambar.
29.00	29.00
Mysore Sada Dosa	Masala dosa is a South Indian crepe made from fermented rice and lentil batter, filled with a spiced potato filling and served with chutneys and sambar.
35.00	35.00
Onion Uttappa	Uttapan is a savory South Indian pancake made from a fermented batter of rice and lentils, typically topped with onions, tomatoes, chilies, and sometimes other vegetables, cooked on a griddle.
33.00	33.00
Poha	Poha is a traditional Indian breakfast dish made from flattened rice, typically cooked with onions, mustard seeds, curry leaves, and spices.
24.00	24.00
Ragda Samosa	Ragda Samosa is a popular Indian street food consisting of crispy samosas served with a spicy curry made from white peas (Ragda), topped with chutneys and garnishes.
35.00	35.00
Sada Dosa	Masala dosa is a South Indian crepe made from fermented rice and lentil batter, filled with a spiced potato filling and served with chutneys and sambar.
23.00	23.00
Sada Uttappa	Uttapan is a savory South Indian pancake made from a fermented batter of rice and lentils, typically topped with onions, tomatoes, chilies, and sometimes other vegetables, cooked on a griddle.
24.00	24.00
Samosa Pav	Samosa pav is a fusion street food where a samosa is sandwiched between a pav bread roll, typically served with chutneys and sometimes garnished with onions and cilantro.
14.00	14.00
Samosa Plate	A samosa plate typically consists of crispy triangular pastry filled with spiced potatoes and peas, served alongside tangy tamarind chutney and minty coriander chutney.
21.00	21.00
Samosa Usal	Samosa Usal is a fusion snack combining the flavors of samosa and pav bhaji, typically featuring samosa pieces mixed with a spicy pav bhaji gravy, served with pav or bread rolls.
35.00	35.00
Schezwan Masala Dosa	Masala dosa is a South Indian crepe made from fermented rice and lentil batter, filled with a spiced potato filling and served with chutneys and sambar.
35.00	35.00
Schezwan Sada Dosa	Masala dosa is a South Indian crepe made from fermented rice and lentil batter, filled with a spiced potato filling and served with chutneys and sambar.
35.00	35.00
Tea	Tea is a hot or cold beverage made by steeping processed leaves, buds, or twigs of the Camellia sinensis plant in hot water, often enjoyed for its soothing flavor and caffeine content.
11.00	11.00
Upma	Upma is a savory South Indian dish made from roasted semolina (rava), cooked with spices, vegetables, and sometimes nuts.
25.00	25.00
Upma Poha	Upma poha is a traditional Indian breakfast dish made from flattened rice (poha) cooked with onions, mustard seeds, curry leaves, and various spices.
28.00	28.00
Vada Pav	Vadapav is a popular Indian street food consisting of a spicy mashed potato patty sandwiched between soft bread rolls, typically served with chutneys and spices.
14.00	14.00
Vada Plate	Vadaplate is a popular Indian street food consisting of spicy mashed potato fritters sandwiched between soft bread rolls, typically served with chutneys and optional toppings.
22.00	22.00
Vada Usal	Vadusas are savory South Indian snacks consisting of crispy fried lentil fritters served with a tangy and spicy lentil stew.

Upma Poha	Upma poha is a traditional Indian breakfast dish made from flattened rice (poha) cooked with onions, mustard seeds, curry leaves, and various spices.
28.00	28.00
Vada Pav	Vadapav is a popular Indian street food consisting of a spicy mashed potato patty sandwiched between soft bread rolls, typically served with chutneys and spices.
14.00	14.00
Vada Plate	Vadaplate is a popular Indian street food consisting of spicy mashed potato fritters sandwiched between soft bread rolls, typically served with chutneys and optional toppings.
22.00	22.00
Vada Usal	Vadusas are savory South Indian snacks consisting of crispy fried lentil fritters served with a tangy and spicy lentil stew.
29.00	29.00
Veg Chesse Sandwich	A veg sandwich typically consists of various vegetables such as lettuce, tomatoes, cucumbers, and bell peppers, layered between slices of bread, often accompanied by spreads like mayonnaise or mustard.
50.00	50.00
Veg Fried Rice	Veg fried rice is a savory dish made by stir-frying cooked rice with mixed vegetables such as carrots, peas, bell peppers, and onions, seasoned with soy sauce and other spices.
50.00	50.00
Veg Hakka Noodles	Vegetable hakka noodles is a stir-fried Indo-Chinese dish featuring noodles cooked with an assortment of vegetables and flavored with soy sauce and other seasonings.
50.00	50.00
Veg Sandwich	A veg sandwich typically consists of various vegetables such as lettuce, tomatoes, cucumbers, and bell peppers, layered between slices of bread, often accompanied by spreads like mayonnaise or mustard.
33.00	33.00
Veg Schezwan Noodles	Vegetable schezwan noodles is a stir-fried Indo-Chinese dish featuring noodles cooked with an assortment of vegetables and flavored with soy sauce and other seasonings.
55.00	55.00
Veg Schezwan Rice	Veg fried rice is a savory dish made by stir-frying cooked rice with mixed vegetables such as carrots, peas, bell peppers, and onions, seasoned with soy sauce and other spices.
55.00	55.00

40 rows in set (0.001 sec) SELECT Item_name, Contents, Price FROM Item;

4.Insert a new order into the Order table.

```
MariaDB [canteen_management]> INSERT INTO `Order` (Order_id, Item_id, Quantity) VALUES (101, 5, 2);
Query OK, 1 row affected (0.004 sec)
```

5.Update the quantity of an existing order in the Order table.

```
MariaDB [canteen_management]> UPDATE `Order` SET Quantity = 3 WHERE Order_id = 101 AND Item_id = 5;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

6.Delete an order from the Order table.

```
MariaDB [canteen_management]> DELETE FROM `Order` WHERE Order_id = 101;
Query OK, 1 row affected (0.003 sec)
```

7. Select all orders with a quantity greater than 2 from the Order table.

```
MariaDB [canteen_management]> SELECT * FROM `Order` WHERE Quantity > 2;
+-----+-----+
| Order_id | Item_id | Quantity |
+-----+-----+
| 323 | 37 | 3 |
| 362 | 40 | 3 |
| 664 | 36 | 3 |
| 700 | 20 | 3 |
| 909 | 30 | 3 |
+-----+-----+
5 rows in set (0.001 sec)
```

8. Select all items with a price between 5 and 10 from the Item table.

```
MariaDB [canteen_management]> SELECT * FROM Item WHERE Price BETWEEN 5 AND 10;
Empty set (0.001 sec)
```

9. Insert a new item into the Item table.

```
MariaDB [canteen_management]> INSERT INTO Item (Item_id, Item_name, Contents, Price) VALUES (51, 'Sandwich', 'Bread, Meat, Vegetables', 7.99);
Query OK, 1 row affected (0.003 sec)
```

10. Update the price of an existing item in the Item table.

```
MariaDB [canteen_management]> UPDATE Item SET Price = 8.99 WHERE Item_id = 51;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

11. Delete an item from the Item table.

```
MariaDB [canteen_management]> DELETE FROM Item WHERE Item_id = 51;
Query OK, 1 row affected (0.002 sec)
```

12. Select all the orders placed by a specific user from the Order_placed_by table.

```
MariaDB [canteen_management]> SELECT * FROM Order_placed_by WHERE User_id = 12;
+-----+-----+
| Order_id | User_id |
+-----+-----+
| 234 | 12 |
+-----+-----+
1 row in set (0.001 sec)
```

13. Insert a new order-user relationship into the Order_placed_by table.

```
MariaDB [canteen_management]> INSERT INTO Order_placed_by (Order_id, User_id) VALUES (710, 15);
Query OK, 1 row affected (0.003 sec)
```

14.Delete an order-user relationship from the Order_placed_by table.

```
MariaDB [canteen_management]> DELETE FROM Order_placed_by WHERE Order_id = 710 AND User_id = 15;
Query OK, 1 row affected (0.004 sec)
```

15.Select all the staff members from the Staff table.

```
MariaDB [canteen_management]> SELECT * FROM Staff;
+-----+-----+
| Staff_id | Staff_Name |
+-----+-----+
|    7891 | Arun      |
|    7894 | Dinesh    |
|    7892 | Kishor    |
|    7895 | Rajesh    |
|    7893 | Swaraj    |
+-----+-----+
5 rows in set (0.001 sec)
```

16.Insert a new staff member into the Staff table.

```
MariaDB [canteen_management]> INSERT INTO Staff (Staff_id, Staff_Name) VALUES (20, 'John Doe');
Query OK, 1 row affected (0.003 sec)
```

17.Update the name of an existing staff member in the Staff table.

```
MariaDB [canteen_management]> UPDATE Staff SET Staff_Name = 'Jane Doe' WHERE Staff_id = 20;
Query OK, 1 row affected (0.002 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

18.Delete a staff member from the Staff table.

```
MariaDB [canteen_management]> DELETE FROM Staff WHERE Staff_id = 20;
Query OK, 1 row affected (0.003 sec)
```

19. Select all the users from the User table.

MariaDB [canteen_management]> SELECT * FROM User;			
User_id	Name	Contact	Address
5	Aaditya	7719865555	Balraj Sahani Marg, Juhu, Mumbai
6	Aastha	7719865556	Sakinaka Junction, Andheri - Kurla Rd, Andheri East, Mumbai
1	Aayush	7719865551	Apollo Bandar, Colaba, Mumbai
7	Abhi	7719865557	Juhu Beach, Juhu, Mumbai, Maharashtra
8	Abhijit	7719865558	Opposite Mittal Industrial Estate, Andheri Kurla Rd, Andheri East, Mumbai
9	Achal	7719865559	Andheri - Kurla Rd, J B Nagar, Andheri East, Mumbai
2	Aditi	7719865552	Dr E Moses Rd, Gandhi Nagar, Upper Worli, Mumbai
3	Amrut	7719865553	Juhu Tara Rd, Uditir Tarang Housing Colony, Juhu Tara, Juhu, Mumbai
10	Anam	7719865550	Chhatrapati Shivaji International Airport, IA Project Rd, Navpada
34	Anish	7620489204	Shop No 18/A Kamdhenu Shopping Centre Lokhandwala
37	Chaitali	7620489207	Shop No.2 Jai Ambe garage, K.N.G Marg, Nr Bank of India, Chembur Naka
11	Dhruv	7620489261	Apollo Bandar, Colaba, Mumbai, Maharashtra 400001, India
12	Gaurav	7620489262	opp. Domestic Airport, Navpada, Vile Parle East, Vile Parle, Mumbai
39	Labdhi	7620489209	Dunhill, Pali Hill Shop 3, Dr Ambedkar Rd, Khar West
13	Maithili	7620489263	Saki Vihar Rd, Murarji Nagar, Mayur Nagar, Passpoli, Powai, Mumbai
14	manas	7620489264	Sahar Airport Road, Andheri - Kurla Rd, near Mumbai International Air
35	Moksh	7620489205	Shop No. 1, Edward Apartment, Off Link Road, Evershine Nagar, Malad West
15	Mrunmayi	7620489265	462, Senapati Bapat Marg, Lower Parel, Mumbai, Maharashtra 400013, India
16	Nishit	7620489266	Plot No.34, 21, MIDC Central Rd, near Akruti Center Point
17	Pavan	7620489267	Oberoi Garden City, International Business Park, Yashodham, Goregaon
38	Pranjal	7620489208	Shop No. 3, Maaz Centre, Opposite The Shop Pali Naka
18	Purvi	7620489268	Nariman Point, Mumbai, Maharashtra 400021, India
19	Raj	7620489269	X - 22, MIDC Central Rd, Hanuman Nagar
20	Riddhi	7620489260	Cambridge School, Karishma Chambers, Off. Sahar Rd

33	Ruchi	7620489203	Marine Mansion, Shop No.110, 1st Marine Street, Anandilal Podar Marg
21	Sachin	7620489201	19A, Sunshine Building, Opp: Domino's Pizza, 1st Cross Road
22	Sahil	7620489202	333, Dr Ambedkar Rd, Pali Pathar, Khar West
23	Sarvesh	7620489203	14/A, SV Rd, next to Khar Masjid, Khar, Khar Danda
24	Shon	7620489204	6, 1st floor, Puran Niwas Bldg. Opp Radio C
25	Shravani	7620489205	Sagar Fortune, Ground Floor, First Floor, Water Field Road
26	Shubham	7620489206	505 , Mastermind-1 , Royal Palm Estate,, Arey Colony
36	Soham	7620489206	Shop No 4, Parvati Bhavan Opp Dr Sanap Clinic, GD Ambedkar Marg
27	Sumit	7620489207	India House No.2, Near Indian Bank Pedder Road
28	Tanmay	7620489208	Unit 39, Madhu Corporate Park, Pandurang Budhkar Marg, Worli, Mumbai
4	Toro	7719865554	Sahar Airport Rd, Ashok Nagar, Andheri East, Mumbai
32	Vaibhavi	7620489202	201, Monterossa, 2nd Floor, 90 Feet Road, Pant Nagar
29	Vansh	7620489209	Shop No. 7, 15th Rd, Kamal Kunj, Bandra West, Mumbai
31	Vedika	7620489211	Bandra Kurla Complex, Grand Hyatt Shopping Plaza
30	Yash	7620489200	P-57, 23rd Rd, Bandra West, Mumbai, Maharashtra 400050, India
40	Yuvraj	7620489210	Rustom & Co, Arsiwala Building, 3rd Ground floor Wodehouse Rd

20.Insert a new user into the User table.

```
MariaDB [canteen_management]> INSERT INTO User (User_id, Name, Contact, Address) VALUES (45, 'Jane Smith', '555-1234', '123 Main St, Anytown USA');
Query OK, 1 row affected (0.003 sec)
```

21.Update the contact information of an existing user in the User table.

```
MariaDB [canteen_management]> UPDATE User SET Contact = '555-5678' WHERE User_id = 45;
Query OK, 1 row affected (0.004 sec)
Rows matched: 1    Changed: 1    Warnings: 0
```

22.Delete a user from the User table.

```
MariaDB [canteen_management]> DELETE FROM User WHERE User_id = 45;
Query OK, 1 row affected (0.004 sec)
```

23.Select all the bills from the Bill table.

```
MariaDB [canteen_management]> SELECT * FROM Bill;
+----+----+----+----+----+
| Bill_id | Amount | Date       | Payment_method | User_id |
+----+----+----+----+----+
| 2200 | 5   | 2024-04-05 | Cash          | 1        |
| 2201 | 8   | 2024-04-05 | Cash          | 2        |
| 2202 | 45  | 2024-04-05 | Cash          | 3        |
| 2203 | 10  | 2024-04-05 | Cash          | 4        |
| 2204 | 10  | 2024-04-05 | Cash          | 5        |
| 2205 | 20  | 2024-04-05 | Cash          | 6        |
| 2206 | 10  | 2024-04-05 | Cash          | 7        |
| 2207 | 20  | 2024-04-05 | Cash          | 8        |
| 2208 | 10  | 2024-04-05 | Cash          | 9        |
| 2209 | 20  | 2024-04-05 | Cash          | 10       |
| 2210 | 40  | 2024-04-05 | Cash          | 11       |
| 2211 | 5   | 2024-04-05 | Cash          | 12       |
| 2212 | 38  | 2024-04-05 | Cash          | 13       |
| 2213 | 40  | 2024-04-05 | Cash          | 14       |
| 2214 | 10  | 2024-04-05 | Cash          | 15       |
| 2215 | 20  | 2024-04-05 | Cash          | 16       |
| 2216 | 10  | 2024-04-05 | Cash          | 17       |
| 2217 | 15  | 2024-04-05 | Cash          | 18       |
| 2218 | 10  | 2024-04-05 | Cash          | 19       |
| 2219 | 30  | 2024-04-05 | Cash          | 20       |
| 2220 | 30  | 2024-04-05 | Cash          | 21       |
| 2221 | 20  | 2024-04-05 | Cash          | 22       |
| 2222 | 25  | 2024-04-05 | Cash          | 23       |
| 2223 | 10  | 2024-04-05 | Cash          | 24       |
| 2224 | 10  | 2024-04-05 | Cash          | 25       |
| 2225 | 20  | 2024-04-05 | Cash          | 26       |
| 2226 | 20  | 2024-04-05 | Cash          | 27       |
| 2227 | 40  | 2024-04-05 | Cash          | 28       |
| 2228 | 28  | 2024-04-05 | Cash          | 29       |
| 2229 | 60  | 2024-04-05 | Cash          | 30       |
| 2230 | 20  | 2024-04-05 | Cash          | 31       |
| 2231 | 20  | 2024-04-05 | Cash          | 32       |
| 2232 | 20  | 2024-04-05 | Cash          | 33       |
| 2233 | 20  | 2024-04-05 | Cash          | 34       |
| 2234 | 10  | 2024-04-05 | Cash          | 35       |
| 2235 | 90  | 2024-04-05 | Cash          | 36       |
| 2236 | 80  | 2024-04-05 | Cash          | 37       |
| 2237 | 10  | 2024-04-05 | Cash          | 38       |
| 2238 | 30  | 2024-04-05 | Cash          | 39       |
| 2239 | 40  | 2024-04-05 | Cash          | 40       |
+----+----+----+----+----+
40 rows in set (0.001 sec)
```

24.Insert a new bill into the Bill table.

```
MariaDB [canteen_management]> INSERT INTO Bill (Bill_id, Amount, Date, Payment_method, User_id) VALUES (908, 50.00, '2023-04-10', 'Credit Card', 12);
Query OK, 1 row affected (0.003 sec)
```

25.Update the payment method of an existing bill in the Bill table.

```
MariaDB [canteen_management]> UPDATE Bill SET Payment_method = 'Cash' WHERE Bill_id = 908;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

26.Delete a bill from the Bill table.

```
MariaDB [canteen_management]> DELETE FROM Bill WHERE Bill_id = 908;
Query OK, 1 row affected (0.003 sec)
```

27.Select all the orders assigned to a specific staff member from the Order_assigned_to table.

```
MariaDB [canteen_management]> SELECT * FROM Order_assigned_to WHERE Staff_id = 7891;
+-----+-----+
| Order_id | Staff_id |
+-----+-----+
|     23   |    7891  |
|    241   |    7891  |
|    255   |    7891  |
|    362   |    7891  |
|    694   |    7891  |
|    717   |    7891  |
|    844   |    7891  |
|    909   |    7891  |
+-----+-----+
8 rows in set (0.001 sec)
```

28.Insert a new order-staff relationship into the Order_assigned_to table.

```
MariaDB [canteen_management]> INSERT INTO Order_assigned_to (Order_id, Staff_id) VALUES (800, 7891);
Query OK, 1 row affected (0.003 sec)
```

29.Delete an order-staff relationship from the Order_assigned_to table.

```
MariaDB [canteen_management]> DELETE FROM Order_assigned_to WHERE Order_id = 800 AND Staff_id = 7891;
Query OK, 1 row affected (0.004 sec)
```

30.Select all the admins from the Admin table.

```
MariaDB [canteen_management]> SELECT * FROM Admin;
+-----+-----+
| Admin_id | Admin_Name |
+-----+-----+
| 80031   | Arun_S
| 80037   | Mukesh
+-----+-----+
2 rows in set (0.001 sec)
```

31.Insert a new admin into the Admin table.

```
ERROR 1054 (42S22): Unknown column 'Name' in 'field list'
MariaDB [canteen_management]> INSERT INTO Admin (Admin_id, Admin_Name) VALUES (30, 'Jane Doe');
Query OK, 1 row affected (0.003 sec)
```

32.Update the name of an existing admin in the Admin table.

```
MariaDB [canteen_management]> UPDATE Admin SET Admin_Name = 'John Doe' WHERE Admin_id = 30;
Query OK, 1 row affected (0.004 sec)
Rows matched: 1    Changed: 1    Warnings: 0
```

33.Delete an admin from the Admin table.

```
MariaDB [canteen_management]> DELETE FROM Admin WHERE Admin_id = 30;
Query OK, 1 row affected (0.003 sec)
```

34.Select all the management relationships from the Admin_manages_staff table.

```
MariaDB [canteen_management]> SELECT * FROM Admin_manages_staff;
+-----+-----+-----+
| Management_key | Staff_id | Admin_id |
+-----+-----+-----+
|      2         | 7891    | 80031   |
|      3         | 7892    | 80037   |
|      4         | 7893    | 80037   |
|      5         | 7894    | 80037   |
|      6         | 7895    | 80031   |
|      7         | 7891    | 80031   |
|      8         | 7895    | 80037   |
|      9         | 7892    | 80037   |
|     10         | 7895    | 80037   |
|     11         | 7893    | 80037   |
|     12         | 7892    | 80037   |
|     13         | 7894    | 80031   |
|     14         | 7895    | 80031   |
|     15         | 7891    | 80031   |
|     16         | 7894    | 80031   |
|     17         | 7893    | 80031   |
|     18         | 7894    | 80031   |
|     19         | 7891    | 80037   |
|     20         | 7895    | 80037   |
|     21         | 7893    | 80037   |
|     22         | 7894    | 80031   |
|     23         | 7891    | 80037   |
|     24         | 7891    | 80037   |
|     25         | 7895    | 80037   |
|     26         | 7892    | 80037   |
|     27         | 7893    | 80037   |
|     28         | 7894    | 80037   |
|     29         | 7894    | 80031   |
|     30         | 7894    | 80031   |
|     31         | 7891    | 80031   |
|     32         | 7895    | 80031   |
|     33         | 7893    | 80031   |
|     34         | 7894    | 80031   |
|     35         | 7892    | 80037   |
|     36         | 7891    | 80037   |
|     37         | 7893    | 80037   |
|     38         | 7892    | 80031   |
|     39         | 7895    | 80037   |
|     40         | 7894    | 80031   |
|     41         | 7891    | 80031   |
+-----+-----+-----+
40 rows in set (0.001 sec)
```

35. Insert a new management relationship into the Admin_manages_staff table.

```
MariaDB [canteen_management]> INSERT INTO Admin_manages_staff (Management_key, Staff_id, Admin_id)
VALUES (42, 7891, 80031);
Query OK, 1 row affected (0.003 sec)
```

36. Delete a management relationship from the Admin_manages_staff table.

```
MariaDB [canteen_management]> DELETE FROM Admin_manages_staff WHERE Management_key = 42;
Query OK, 1 row affected (0.003 sec)
```

37. Select all the items that are part of a specific menu from the Menu table.

```
MariaDB [canteen_management]> SELECT Item_id FROM Menu WHERE Menu_id = 2210;
+-----+
| Item_id |
+-----+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
| 7 |
| 8 |
| 9 |
| 10 |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |
| 21 |
| 22 |
| 23 |
| 24 |
| 25 |
| 26 |
| 27 |
| 28 |
| 29 |
| 30 |
| 31 |
| 32 |
| 33 |
| 34 |
| 35 |
| 36 |
| 37 |
| 38 |
| 39 |
| 40 |
+-----+
40 rows in set (0.001 sec)
```

38.Insert a new menu-item relationship into the Menu table.

```
MariaDB [canteen_management]> INSERT INTO Menu (Menu_id, Item_id) VALUES (2210, 45);
Query OK, 1 row affected (0.003 sec)
```

39.Delete a menu-item relationship from the Menu table.

```
MariaDB [canteen_management]> DELETE FROM Menu WHERE Menu_id = 2210 AND Item_id = 45;
Query OK, 1 row affected (0.002 sec)
```

40.Select all the orders with a specific quantity from the Order table.

```
MariaDB [canteen_management]> SELECT * FROM `Order` WHERE Quantity = 3;
+-----+-----+-----+
| Order_id | Item_id | Quantity |
+-----+-----+-----+
|     323 |      37 |        3 |
|     362 |      40 |        3 |
|     664 |      36 |        3 |
|     700 |      20 |        3 |
|     909 |      30 |        3 |
+-----+-----+-----+
5 rows in set (0.001 sec)
```

41.Select all the items with a specific name, contents, or price from the Item table.

```
MariaDB [canteen_management]> SELECT * FROM Item WHERE Item_name = 'Veg Sandwich' OR Contents = 'Vadaplate is a popular Indian street food consisting of spicy mashed potato fritters sandwiched between soft bread rolls, typically served with chutneys and optional toppings.' OR Price = 24;
+-----+-----+
| Item_id | Item_name      | Contents
|          |                  | Price |
+-----+-----+
|       6 | Idli-Sambhar | Idli sambar is a traditional South Indian dish consisting of steamed rice cakes (idli) served with a flavorful lentil-based stew (sambar).
|          |                  | 24.00 |
|       5 | Poha           | Poha is a traditional Indian breakfast dish made from flattened rice, typically cooked with onions, mustard seeds, curry leaves, and spices.
|          |                  | 24.00 |
|      32 | Sada Uttappa | Uttapam is a savory South Indian pancake made from a fermented batter of rice and lentils, typically topped with onions, tomatoes, chilies, and sometimes other vegetables, cooked on a griddle.
|          |                  | 24.00 |
|       4 | Samosa Plate | A samosa plate typically consists of crispy triangular pastry filled with spiced potatoes and peas, served alongside tangy tamarind chutney and minty coriander chutney.
|          |                  | 24.00 |
|       3 | Vada Plate   | Vadaplate is a popular Indian street food consisting of spicy mashed potato fritters sandwiched between soft bread rolls, typically served with chutneys and optional toppings.
|          |                  | 22.00 |
|      15 | Veg Sandwich | A veg sandwich typically consists of various vegetables such as lettuce, tomatoes, cucumbers, and bell peppers, layered between slices of bread, often accompanied by spreads like mayonnaise or mustard.
|          |                  | 33.00 |
+-----+-----+
6 rows in set (0.001 sec)
```

42. Select all the orders placed by a specific user and assigned to a specific staff member.

```
MariaDB [canteen_management]> SELECT * FROM Order_placed_by op
    -> JOIN Order_assigned_to oa ON op.Order_id = oa.Order_id
    -> WHERE op.User_id = 1 AND oa.Staff_id = 7891;
+-----+-----+-----+-----+
| Order_id | User_id | Order_id | Staff_id |
+-----+-----+-----+-----+
|      241 |       1 |      241 |     7891 |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

43. Select all the bills for a specific user from the Bill table.

```
MariaDB [canteen_management]> SELECT * FROM Bill WHERE User_id = 1;
+-----+-----+-----+-----+
| Bill_id | Amount | Date       | Payment_method | User_id |
+-----+-----+-----+-----+
|    2200 |      5 | 2024-04-05 | Cash           |       1 |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

44. Select all the orders for a specific item from the Order_Item table.

```
MariaDB [canteen_management]> SELECT * FROM Order_Item WHERE Item_id = 5;
+-----+-----+-----+
| Order_id | Item_id | Quantity |
+-----+-----+-----+
|      324 |       5 |        1 |
+-----+-----+-----+
1 row in set (0.001 sec)
```

45. Select the total number of orders for a specific item from the Order_Item table.

```
MariaDB [canteen_management]> SELECT COUNT(*) FROM Order_Item WHERE Item_id = 5;
+-----+
| COUNT(*) |
+-----+
|      1 |
+-----+
1 row in set (0.001 sec)
```

46. Select the total revenue for a specific date range from the Bill table.

```
MariaDB [canteen_management]> SELECT SUM(Amount) AS Total_Revenue
    -> FROM Bill
    -> WHERE Date BETWEEN '2023-01-01' AND '2023-12-31';
+-----+
| Total_Revenue |
+-----+
|        50 |
+-----+
1 row in set (0.001 sec)
```

47.Select the top 10 most ordered items from the Order_Item table.

```
MariaDB [canteen_management]> SELECT Item_id, COUNT(*) AS Total_Orders
-> FROM Order_Item
-> GROUP BY Item_id
-> ORDER BY Total_Orders DESC
-> LIMIT 10;
+-----+
| Item_id | Total_Orders |
+-----+
|      16 |          1 |
|      32 |          1 |
|       1 |          1 |
|      17 |          1 |
|      33 |          1 |
|       2 |          1 |
|      18 |          1 |
|      34 |          1 |
|       3 |          1 |
|      19 |          1 |
+-----+
10 rows in set (0.001 sec)
```

48.Select the average order quantity for a specific item from the Order_Item table.

```
MariaDB [canteen_management]> SELECT AVG(Quantity) AS Avg_Quantity
-> FROM Order_Item
-> WHERE Item_id = 5;
+-----+
| Avg_Quantity |
+-----+
|     1.0000 |
+-----+
1 row in set (0.001 sec)
```

49.Select the total number of orders placed by a specific user from the Order_placed_by table.

```
MariaDB [canteen_management]> SELECT COUNT(*) AS Total_Orders
-> FROM Order_placed_by
-> WHERE User_id = 12;
+-----+
| Total_Orders |
+-----+
|       1 |
+-----+
1 row in set (0.001 sec)
```

50.Select the total number of staff members managed by a specific admin from the Admin_manages_staff table.

```
MariaDB [canteen_management]> SELECT COUNT(*) AS Total_Staff
-> FROM Admin_manages_staff
-> WHERE Admin_id = 15;
+-----+
| Total_Staff |
+-----+
|          0 |
+-----+
1 row in set (0.001 sec)
```

51.Select the total number of bills paid by a specific payment method from the Bill table.

```
MariaDB [canteen_management]> SELECT Payment_method, COUNT(*) AS Total_Bills
-> FROM Bill
-> GROUP BY Payment_method;
+-----+-----+
| Payment_method | Total_Bills |
+-----+-----+
| Cash           |        40 |
| Credit Card    |         1 |
+-----+-----+
2 rows in set (0.001 sec)
```

52.Select the top 5 users with the highest total bill amounts from the Bill table.

```
MariaDB [canteen_management]> SELECT User_id, SUM(Amount) AS Total_Bill_Amount
-> FROM Bill
-> GROUP BY User_id
-> ORDER BY Total_Bill_Amount DESC
-> LIMIT 5;
+-----+
| User_id | Total_Bill_Amount |
+-----+
|   36   |        90 |
|   37   |        80 |
|   30   |        60 |
|   12   |        55 |
|    3   |        45 |
+-----+
5 rows in set (0.001 sec)
```

53.Select the total number of orders placed on a specific date from the Order table.

```
MariaDB [canteen_management]> SELECT COUNT(*) AS Total_Orders
->
-> FROM `Order`
->
-> WHERE Order_id IN (
->
->   SELECT Order_id
->
->   FROM Order_placed_by
->
->   WHERE Order_id IN (
->
->     SELECT Order_id
->
->     FROM Bill
->
->     WHERE Date = '2023-04-10'
->
->   )
->
-> );
+-----+
| Total_Orders |
+-----+
|          40 |
+-----+
1 row in set (0.001 sec)
```

54.Select the total number of unique items ordered from the Order_Item table.

```
MariaDB [canteen_management]> SELECT COUNT(DISTINCT Item_id) AS Total_Unique_Items
-> FROM Order_Item;
+-----+
| Total_Unique_Items |
+-----+
|          40 |
+-----+
1 row in set (0.001 sec)
```

55.Select the top 3 most popular menu items based on total orders from the Order_Item table.

```
MariaDB [canteen_management]> SELECT i.Item_name, COUNT(*) AS Total_Orders
-> FROM Order_Item oi
-> JOIN Item i ON oi.Item_id = i.Item_id
-> GROUP BY i.Item_name
-> ORDER BY Total_Orders DESC
-> LIMIT 3;
+-----+-----+
| Item_name | Total_Orders |
+-----+-----+
| Vada Pav   |          1 |
| Samosa Pav |          1 |
| Vada Plate |          1 |
+-----+-----+
3 rows in set (0.001 sec)
```

56.Calculating Average Price of Items:

```
MariaDB [canteen_management]> SELECT AVG(Price) AS Avg_Price FROM Item;
+-----+
| Avg_Price |
+-----+
| 32.450000 |
+-----+
1 row in set (0.001 sec)
```

57.Calculating Maximum Quantity Ordered:

```
MariaDB [canteen_management]> SELECT MAX(Quantity) AS Max_Quantity FROM Order_Item;
+-----+
| Max_Quantity |
+-----+
|          3 |
+-----+
1 row in set (0.001 sec)
```

58.Calculating Maximum Price of Items:

```
MariaDB [canteen_management]> SELECT MAX(Price) AS Max_Price FROM Item;
+-----+
| Max_Price |
+-----+
|     64.00 |
+-----+
1 row in set (0.001 sec)
```

59.Calculating Minimum Quantity Ordered:

```
MariaDB [canteen_management]> SELECT MIN(Quantity) AS Min_Quantity FROM Order_Item;
+-----+
| Min_Quantity |
+-----+
|          1   |
+-----+
1 row in set (0.001 sec)
```

60.Calculating Minimum Price of Items:

```
MariaDB [canteen_management]> SELECT MIN(Price) AS Min_Price FROM Item;
+-----+
| Min_Price |
+-----+
|     11.00  |
+-----+
1 row in set (0.001 sec)
```

61.Finding Total Orders per User:

```
MariaDB [canteen_management]> SELECT User_id, COUNT(*) AS Total_Orders FROM Order_placed_by GROUP
BY User_id;
+-----+-----+
| User_id | Total_Orders |
+-----+-----+
|      1  |          1 |
|      2  |          1 |
|      3  |          1 |
|      4  |          1 |
|      5  |          1 |
|      6  |          1 |
|      7  |          1 |
|      8  |          1 |
|      9  |          1 |
|     10  |          1 |
|     11  |          1 |
|     12  |          1 |
|     13  |          1 |
|     14  |          1 |
|     15  |          1 |
|     16  |          1 |
|     17  |          1 |
|     18  |          1 |
|     19  |          1 |
|     20  |          1 |
|     21  |          1 |
|     22  |          1 |
|     23  |          1 |
|     24  |          1 |
|     25  |          1 |
|     26  |          1 |
|     27  |          1 |
|     28  |          1 |
|     29  |          1 |
|     30  |          1 |
|     31  |          1 |
|     32  |          1 |
|     33  |          1 |
|     34  |          1 |
|     35  |          1 |
|     36  |          1 |
|     37  |          1 |
|     38  |          1 |
|     39  |          1 |
|     40  |          1 |
+-----+-----+
40 rows in set (0.001 sec)
```

62.Finding Total Amount Paid per User:

```
MariaDB [canteen_management]> SELECT user_id, SUM(Amount) AS Total_Amount FROM Bill GROUP BY user_id;
```

User_id	Total_Amount
1	5
2	8
3	45
4	10
5	10
6	20
7	10
8	20
9	10
10	20
11	40
12	55
13	38
14	40
15	10
16	20
17	10
18	15
19	10
20	30
21	30
22	20
23	25
24	10
25	10
26	20
27	20
28	40
29	28
30	60
31	20
32	20
33	20
34	20
35	10
36	90
37	80
38	10
39	30
40	40

63.Finding Staff Assigned Orders:

```
MariaDB [canteen_management]> SELECT Staff_id, COUNT(*) AS Total_Orders FROM Order_assigned_to GROUP BY Staff_id;
```

Staff_id	Total_Orders
7891	8
7892	6
7893	7
7894	11
7895	8

5 rows in set (0.001 sec)

64.Finding Total Quantity Sold per Item:

```
MariaDB [canteen_management]> SELECT Item_id, SUM(Quantity) AS Total_Quantity FROM Order_Item GROUP BY Item_id;
```

Item_id	Total_Quantity
1	1
2	2
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	2
12	1
13	2
14	1
15	1
16	1
17	1
18	1
19	1
20	3
21	1
22	1
23	1
24	1
25	1
26	1
27	1
28	2
29	1
30	3
31	1
32	2
33	1
34	1
35	1
36	3
37	3
38	1
39	1
40	3

40 rows in set (0.001 sec)

65.Finding Total Sales per Day:

```
MariaDB [canteen_management]> SELECT Date, SUM(Amount) AS Total_Sales FROM Bill GROUP BY Date;
+-----+-----+
| Date      | Total_Sales |
+-----+-----+
| 2023-04-10 |        50 |
| 2024-04-05 |      979 |
+-----+-----+
2 rows in set (0.001 sec)
```

66.Finding Total Sales per Payment Method:

```
MariaDB [canteen_management]> SELECT Payment_method, SUM(Amount) AS Total_Sales FROM Bill GROUP BY Payment_method;
+-----+-----+
| Payment_method | Total_Sales |
+-----+-----+
| Cash          |      979 |
| Credit Card   |       50 |
+-----+-----+
2 rows in set (0.001 sec)
```

67.Finding Highest Earning User:

```
MariaDB [canteen_management]> SELECT User_id, SUM(Amount) AS Total_Amount FROM Bill GROUP BY User_id ORDER BY Total_Amount DESC LIMIT 1;
+-----+-----+
| User_id | Total_Amount |
+-----+-----+
|      36 |        90 |
+-----+-----+
1 row in set (0.001 sec)
```

68.Finding Most Active Staff Member:

```
MariaDB [canteen_management]> SELECT Staff_id, COUNT(*) AS Total_Orders FROM Order_assigned_to GROUP BY Staff_id ORDER BY Total_Orders DESC LIMIT 1;
+-----+-----+
| Staff_id | Total_Orders |
+-----+-----+
|    7894 |        11 |
+-----+-----+
1 row in set (0.001 sec)
```

69.Finding Total Amount Spent in a Day:

```
MariaDB [canteen_management]> SELECT Date, SUM(Amount) AS Total_Amount FROM Bill GROUP BY Date;
+-----+-----+
| Date      | Total_Amount |
+-----+-----+
| 2023-04-10 |        50 |
| 2024-04-05 |      979 |
+-----+-----+
2 rows in set (0.001 sec)
```

70.Finding Total Quantity Sold by a Staff Member:

```
MariaDB [canteen_management]> SELECT Staff_id, SUM(Quantity) AS Total_Quantity FROM Order_assigned_to JOIN Order_Item ON Order_assigned_to.Order_id = Order_Item.Order_id GROUP BY Staff_id;
+-----+-----+
| Staff_id | Total_Quantity |
+-----+-----+
|    7891   |        12 |
|    7892   |        10 |
|    7893   |        12 |
|    7894   |        12 |
|    7895   |         9 |
+-----+-----+
5 rows in set (0.001 sec)
```

71.Finding Total Amount Earned by a Staff Member:

```
MariaDB [canteen_management]> SELECT Staff_id, SUM(Price * Quantity) AS Total_Earnings FROM Order_assigned_to JOIN Order_Item ON Order_assigned_to.Order_id = Order_Item.Order_id JOIN Item ON Order_Item.Item_id = Item.Item_id GROUP BY Staff_id;
+-----+-----+
| Staff_id | Total_Earnings |
+-----+-----+
|    7891   |     400.00 |
|    7892   |     217.00 |
|    7893   |     520.00 |
|    7894   |     402.00 |
|    7895   |     260.00 |
+-----+-----+
5 rows in set (0.001 sec)
```

72.Finding Total Quantity Sold of an Item by a Staff Member:

```
MariaDB [canteen_management]> SELECT Staff_id, Item_id, SUM(Quantity) AS Total_Quantity FROM Order_assigned_to JOIN Order_Item ON Order_assigned_to.Order_id = Order_Item.Order_id GROUP BY Staff_id, Item_id;
+-----+-----+-----+
| Staff_id | Item_id | Total_Quantity |
+-----+-----+-----+
|    7891   |      1   |        1 |
|    7891   |      6   |        1 |
|    7891   |     14   |        1 |
|    7891   |     18   |        1 |
|    7891   |     22   |        1 |
|    7891   |     30   |        3 |
|    7891   |     35   |        1 |
|    7891   |     40   |        3 |
|    7892   |      2   |        2 |
|    7892   |      8   |        1 |
|    7892   |     11   |        2 |
|    7892   |     25   |        1 |
|    7892   |     34   |        1 |
|    7892   |     37   |        3 |
|    7893   |      3   |        1 |
|    7893   |     10   |        1 |
|    7893   |     16   |        1 |
|    7893   |     20   |        3 |
|    7893   |     26   |        1 |
|    7893   |     32   |        2 |
|    7893   |     36   |        3 |
|    7894   |      4   |        1 |
|    7894   |     12   |        1 |
|    7894   |     15   |        1 |
|    7894   |     17   |        1 |
|    7894   |     21   |        1 |
|    7894   |     22   |        1 |
|    7894   |     27   |        1 |
|    7894   |     28   |        2 |
|    7894   |     29   |        1 |
|    7894   |     33   |        1 |
|    7894   |     39   |        1 |
|    7895   |      5   |        1 |
|    7895   |      7   |        1 |
|    7895   |      9   |        1 |
|    7895   |     13   |        2 |
|    7895   |     19   |        1 |
|    7895   |     24   |        1 |
|    7895   |     31   |        1 |
|    7895   |     38   |        1 |
+-----+-----+-----+
40 rows in set (0.001 sec)
```

73. Select the total number of orders placed by each user from the Order_placed_by table.

```
MariaDB [canteen_management]> SELECT u.Name, COUNT(*) AS Total_Orders
-> FROM Order_Placed_By op
-> JOIN User u ON op.User_id = u.User_id
-> GROUP BY u.Name;
+-----+-----+
| Name | Total_Orders |
+-----+-----+
| Aaditya | 1 |
| Aastha | 1 |
| Aayush | 1 |
| Abhi | 1 |
| Abhijit | 1 |
| Achal | 1 |
| Aditi | 1 |
| Amrut | 1 |
| Anam | 1 |
| Anish | 1 |
| Chaitali | 1 |
| Dhruv | 1 |
| Gaurav | 1 |
| Labdh | 1 |
| Maithili | 1 |
| manas | 1 |
| Moksh | 1 |
| Mrunmayi | 1 |
| Nishit | 1 |
| Pavan | 1 |
| Pranjal | 1 |
| Purvi | 1 |
| Raj | 1 |
| Riddhi | 1 |
| Ruchi | 1 |
| Sachin | 1 |
| Sahil | 1 |
| Sarvesh | 1 |
| Shon | 1 |
| Shravani | 1 |
| Shubham | 1 |
| Soham | 1 |
| Sumit | 1 |
| Tanmay | 1 |
| Toro | 1 |
| Vaibhavi | 1 |
| Vansh | 1 |
| Vedika | 1 |
| Yash | 1 |
| Yuvraj | 1 |
+-----+-----+
40 rows in set (0.001 sec)
```

74. Select the total revenue generated by each payment method from the Bill table.

```
MariaDB [canteen_management]> SELECT Payment_Method, SUM(Amount) AS Total_Revenue
-> FROM Bill
-> GROUP BY Payment_Method;
+-----+-----+
| Payment_Method | Total_Revenue |
+-----+-----+
| Cash | 979 |
| Credit Card | 50 |
+-----+-----+
2 rows in set (0.001 sec)
```

75. Select the top 5 items with the highest total orders from the Order_Item table.

```
ERROR 1054 (42S22): Unknown column 'oi.Order_id' in 'on clause'
MariaDB [canteen_management]> SELECT i.Item_name, COUNT(*) AS Total_Orders
-> FROM Order_Item oi
-> JOIN Item i ON oi.Item_id = i.Item_id
-> GROUP BY i.Item_name
-> ORDER BY Total_Orders DESC
-> LIMIT 5;
+-----+-----+
| Item_name | Total_Orders |
+-----+-----+
| Vada Pav | 1 |
| Samosa Pav | 1 |
| Vada Plate | 1 |
| Samosa Plate | 1 |
| Poha | 1 |
+-----+-----+
5 rows in set (0.001 sec)
```

76. Select the total revenue generated by each payment method, the number of bills paid by each method, and the average bill amount for each payment method from the Bill table.

```
MariaDB [canteen_management]> SELECT Payment_method, SUM(Amount) AS Total_Revenue, COUNT(*) AS Total_Bills, AVG(Amount) AS Avg_Bill_Amount
    -> FROM Bill
    -> GROUP BY Payment_method;
+-----+-----+-----+-----+
| Payment_method | Total_Revenue | Total_Bills | Avg_Bill_Amount |
+-----+-----+-----+
| Cash          |      979     |       40    |      24.4750   |
| Credit Card   |       50     |        1    |      50.0000   |
+-----+-----+-----+
2 rows in set (0.001 sec)
```

77. Select the top 10 users with the highest total bill amounts and the percentage of total revenue that their orders contribute to.

```
ERROR 1054 (42S22): Unknown column 'b.order_id' in 'on clause'
MariaDB [canteen_management]> SELECT u.Name, SUM(b.Amount) AS Total_Bill_Amount,
    ->           (SUM(b.Amount) / (SELECT SUM(Amount) FROM Bill)) * 100 AS Percentage_of_Total_Revenue
    -> FROM User u
    -> JOIN Bill b ON u.User_id = b.User_id
    -> GROUP BY u.Name
    -> ORDER BY Total_Bill_Amount DESC
    -> LIMIT 10;
+-----+-----+-----+
| Name      | Total_Bill_Amount | Percentage_of_Total_Revenue |
+-----+-----+-----+
| Soham     |        90         |      8.7464                |
| Chaitali  |        80         |      7.7745                |
| Yash      |        60         |      5.8309                |
| Gaurav    |        55         |      5.3450                |
| Amruth   |        45         |      4.3732                |
| Tanmay   |        40         |      3.8873                |
| manas    |        40         |      3.8873                |
| Dhruv    |        40         |      3.8873                |
| Yuvraj   |        40         |      3.8873                |
| Maithili |        38         |      3.6929                |
+-----+-----+-----+
10 rows in set (0.001 sec)
```

78. Select the top 5 items with the highest average order quantity from the Order_Item table.

```
MariaDB [canteen_management]> SELECT i.Item_name, AVG(oi.Quantity) AS Avg_Quantity
    -> FROM Order_Item oi
    -> JOIN Item i ON oi.Item_id = i.Item_id
    -> GROUP BY i.Item_name
    -> ORDER BY Avg_Quantity DESC
    -> LIMIT 5;
+-----+-----+
| Item_name | Avg_Quantity |
+-----+-----+
| Tea       |      3.0000   |
| Lassi    |      3.0000   |
| Lunch    |      3.0000   |
| Veg Fried Rice | 3.0000 |
| Schezwan Sada Dosa | 3.0000 |
+-----+-----+
5 rows in set (0.001 sec)
```

79. Select the total revenue generated by each payment method, the number of bills paid by each method, and the top 3 payment methods by total revenue.

```
MariaDB [canteen_management]> SELECT Payment_method, SUM(Amount) AS Total_Revenue, COUNT(*) AS Total_Bills
-> FROM Bill
-> GROUP BY Payment_method
-> ORDER BY Total_Revenue DESC
-> LIMIT 3;
+-----+-----+-----+
| Payment_method | Total_Revenue | Total_Bills |
+-----+-----+-----+
| Cash          |      979       |      40      |
| Credit Card   |       50        |       1       |
+-----+-----+-----+
2 rows in set (0.001 sec)
```

80. Select the total number of orders placed by each user, the total revenue generated from those orders, and the top 5 users by total revenue.

```
MariaDB [canteen_management]> SELECT u.Name, COUNT(*) AS Total_Orders, SUM(b.Amount) AS Total_Revenue
-> FROM Order_placed_by op
-> JOIN User u ON op.User_id = u.User_id
-> JOIN Bill b ON op.Order_id = b.Bill_id
-> GROUP BY u.Name
-> ORDER BY Total_Revenue DESC
-> LIMIT 2;
Empty set (0.000 sec)
```

81. Select the top 3 payment methods with the highest average bill amount.

```
ERROR at line 4
MariaDB [canteen_management]> SELECT Payment_method, AVG(Amount) AS Avg_Bill_Amount
-> FROM Bill
-> GROUP BY Payment_method
-> ORDER BY Avg_Bill_Amount DESC
-> LIMIT 3;
+-----+-----+
| Payment_method | Avg_Bill_Amount |
+-----+-----+
| Credit Card   |      50.0000    |
| Cash          |      24.4750    |
+-----+-----+
2 rows in set (0.000 sec)
```

82. Insert a new bill into the Bill table.

```
ERROR 1054 (42S22): Unknown column 'b.Order_id' in 'on clause'
MariaDB [canteen_management]> INSERT INTO Bill (Bill_id, Amount, Date, Payment_method, User_id)
VALUES (708, 50.00, '2023-04-10', 'Credit Card', 12);
Query OK, 1 row affected (0.001 sec)
```

83. Update the payment method of an existing bill in the Bill table.

```
MariaDB [canteen_management]> UPDATE Bill SET Payment_method = 'Cash' WHERE Bill_id = 201;
Query OK, 1 row affected (0.002 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

84.Delete a bill from the Bill table.

```
MariaDB [canteen_management]> DELETE FROM Bill WHERE Bill_id = 708;
Query OK, 1 row affected (0.002 sec)
```

85.Select all the orders assigned to a specific staff member from the Order_assigned_to table.

```
MariaDB [canteen_management]> SELECT * FROM Order_assigned_to WHERE Staff_id = 7893;
+-----+-----+
| Order_id | Staff_id |
+-----+-----+
|      59 |    7893 |
|     444 |    7893 |
|     664 |    7893 |
|     700 |    7893 |
|     710 |    7893 |
|     731 |    7893 |
|    939 |    7893 |
+-----+-----+
7 rows in set (0.000 sec)
```

86.Insert a new admin into the Admin table.

```
MariaDB [canteen_management]> INSERT INTO Admin (Admin_id, Admin_Name) VALUES (95, 'Jane Doe');
Query OK, 1 row affected (0.001 sec)
```

87.Update the name of an existing admin in the Admin table.

```
MariaDB [canteen_management]> UPDATE Admin SET Admin_Name = 'John Doe' WHERE Admin_id = 95;
Query OK, 1 row affected (0.002 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

88.Delete an admin from the Admin table.

```
MariaDB [canteen_management]> DELETE FROM Admin WHERE Admin_id = 30;
Query OK, 0 rows affected (0.000 sec)
```

89.Insert a new menu-item relationship into the Menu table.

```
+-----+-----+
| REFERENCES `Staff` (`Staff_id`) |
MariaDB [canteen_management]> INSERT INTO Menu (Menu_id, Item_id) VALUES (2210, 120);
Query OK, 1 row affected (0.002 sec)
```

90.Delete a menu-item relationship from the Menu table.

```
MariaDB [canteen_management]> DELETE FROM Menu WHERE Menu_id = 2210 AND Item_id = 120;
Query OK, 1 row affected (0.002 sec)
```

91.Select all the orders with a specific quantity from the Order table.

```
MariaDB [canteen_management]> SELECT * FROM `Order` WHERE Quantity = 3;
+-----+-----+-----+
| Order_id | Item_id | Quantity |
+-----+-----+-----+
|     323 |     37 |      3 |
|     362 |     40 |      3 |
|     664 |     36 |      3 |
|     700 |     28 |      3 |
|     989 |     38 |      3 |
+-----+-----+-----+
5 rows in set (0.000 sec)
```

92.Select all the orders for a specific item from the Order_Item table.

```
MariaDB [canteen_management]> SELECT * FROM Order_Item WHERE Item_id = 6;
+-----+-----+-----+
| Order_id | Item_id | Quantity |
+-----+-----+-----+
|      255 |       6 |        1 |
+-----+-----+
```

93.Select the total number of orders for a specific item from the Order_Item table.

```
MariaDB [canteen_management]> SELECT COUNT(*) FROM Order_Item WHERE Item_id = 6;
+-----+
| COUNT(*) |
+-----+
|      1 |
+-----+
1 row in set (0.000 sec)
```

94.Select the total revenue for a specific date range from the Bill table.

```
MariaDB [canteen_management]> SELECT SUM(Amount) AS Total_Revenue
    -> FROM Bill
    -> WHERE Date BETWEEN '2023-01-01' AND '2023-12-31';
+-----+
| Total_Revenue |
+-----+
|        50 |
+-----+
1 row in set (0.000 sec)
```

95.Select the total revenue for a specific date range from the Bill table.

```
MariaDB [canteen_management]> SELECT SUM(Amount) AS Total_Revenue
    -> FROM Bill
    -> WHERE Date BETWEEN '2023-01-01' AND '2023-12-31';
+-----+
| Total_Revenue |
+-----+
|        50 |
+-----+
1 row in set (0.000 sec)
```

96. Retrieve all orders along with the corresponding user's name and contact information.

Order_id	Name	Contact
13	Toro	7719865554
20	Achal	7719865559
23	manas	7620489264
59	Amrut	7719865553
67	Abhijit	7719865558
74	Maithili	7620489263
123	Aditi	7719865552
215	Abhi	7719865557
234	Gaurav	7620489262
241	Aayush	7719865551
255	Aastha	7719865555
288	Dhruv	7620489261
323	Chaitali	7620489267
324	Aaditya	7719865555
331	Mrunmayi	7620489265
362	Yuvraj	7620489210
444	Anam	7719865550
523	Labdhii	7620489269
652	Pranjal	7620489268
661	Sahil	7620489262
663	Sachin	7620489261
664	Soham	7620489260
670	Vansh	7620489269
672	Tanmay	7620489268
694	Moksh	7620489265
700	Riddhi	7620489266
703	Sumit	7620489267
704	Anish	7620489264
710	Nishit	7620489266
713	Raj	7620489269
717	Purvi	7620489268
721	Pavan	7620489267
731	Shubham	7620489268
800	Shravani	7620489265
844	Sarvesh	7620489263
849	Shon	7620489264
878	Ruchi	7620489263
879	Vedika	7620489211
969	Yash	7620489266
939	Vaibhavi	7620489262

97. Retrieve the total revenue generated from all orders placed.

Total_revenue
1799.00
1 row in set (0.000 sec)

98. Retrieve the average amount of each bill.

Average_bill_amount
25.0976
1 row in set (0.000 sec)

99. Retrieve the total number of items sold for each menu item.

```
MariaDB [canteen_management]> SELECT i.Item_name, SUM(oi.Quantity) AS Total_items_sold
-> FROM Item i
-> LEFT JOIN Order_Item oi ON i.Item_id = oi.Item_id
-> GROUP BY i.Item_id, i.Item_name;
+-----+-----+
| Item_name | Total_items_sold |
+-----+-----+
| Vada Pav | 1 |
| Samosa Pav | 2 |
| Vada Plate | 1 |
| Samosa Plate | 1 |
| Poha | 1 |
| Idli-Sambhar | 1 |
| Upma | 1 |
| Upma Poha | 1 |
| Medu Wada | 1 |
| Batata Wada | 1 |
| Misal Pav | 2 |
| Vada Usal | 1 |
| Ragada Samosa | 2 |
| Samosa Usal | 1 |
| Veg Sandwich | 1 |
| Veg Chesse Sandwich | 1 |
| Mumbai Sandwich | 1 |
| Mumbai Cheese Sandwich | 1 |
| Cheese Sandwich | 1 |
| Veg Fried Rice | 3 |
| Veg Schezwan Rice | 1 |
| Veg Hakka Noddles | 1 |
| Veg Schezwan Noodles | 1 |
| Sada Dosa | 1 |
| Masala Dosa | 1 |
| Butter Sada Dosa | 1 |
| Butter Masala Dosa | 1 |
| Mysore Masala Dosa | 2 |
| Mysore Sada Dosa | 1 |
| Schezwan Sada Dosa | 3 |
| Schezwan Masala Dosa | 1 |
| Sada Uttappa | 2 |
| Onion Uttappa | 1 |
| Masala Uttappa | 1 |
| Chinese Dosa | 1 |
| Lunch | 3 |
| Tea | 3 |
| Coffe | 1 |
| Chass | 1 |
| Lassi | 3 |
+-----+-----+
40 rows in set (0.001 sec)
```

100. Retrieve the average quantity of each item ordered.

```
MariaDB [canteen_management]> SELECT i.Item_name, AVG(oi.Quantity) AS Average_quantity_ordered
-> FROM Item i
-> LEFT JOIN Order_Item oi ON i.Item_id = oi.Item_id
-> GROUP BY i.Item_id, i.Item_name;
+-----+-----+
| Item_name | Average_quantity_ordered |
+-----+-----+
| Vada Pav | 1.0000 |
| Samosa Pav | 2.0000 |
| Vada Plate | 1.0000 |
| Samosa Plate | 1.0000 |
| Poha | 1.0000 |
| Idli-Sambhar | 1.0000 |
| Upma | 1.0000 |
| Upma Poha | 1.0000 |
| Medu Wada | 1.0000 |
| Batata Wada | 1.0000 |
| Misal Pav | 2.0000 |
| Vada Usal | 1.0000 |
| Ragada Samosa | 2.0000 |
| Samosa Usal | 1.0000 |
| Veg Sandwich | 1.0000 |
| Veg Chesse Sandwich | 1.0000 |
| Mumbai Sandwich | 1.0000 |
| Mumbai Cheese Sandwich | 1.0000 |
| Cheese Sandwich | 1.0000 |
| Veg Fried Rice | 3.0000 |
| Veg Schezwan Rice | 1.0000 |
| Veg Hakka Noddles | 1.0000 |
| Veg Schezwan Noodles | 1.0000 |
| Sada Dosa | 1.0000 |
| Masala Dosa | 1.0000 |
| Butter Sada Dosa | 1.0000 |
| Butter Masala Dosa | 1.0000 |
| Mysore Masala Dosa | 2.0000 |
| Mysore Sada Dosa | 1.0000 |
| Schezwan Sada Dosa | 3.0000 |
| Schezwan Masala Dosa | 1.0000 |
| Sada Uttappa | 2.0000 |
| Onion Uttappa | 1.0000 |
| Masala Uttappa | 1.0000 |
| Chinese Dosa | 1.0000 |
| Lunch | 3.0000 |
| Tea | 3.0000 |
| Coffe | 1.0000 |
| Chass | 1.0000 |
| Lassi | 3.0000 |
+-----+-----+
40 rows in set (0.001 sec)
```

Conclusion:

We have concluded our experiment in the following manner:

1. **Normalization of the schema to 3NF/BCNF:** Normalizing the schema to the third normal form (3NF) or Boyce-Codd normal form (BCNF) is a crucial step in database design. It ensures that the tables are free from insertion, update, and deletion anomalies, reducing data redundancy and maintaining data integrity. By following the normalization process, we have eliminated partial and transitive dependencies, resulting in a well-structured and efficient database schema.
2. **Preparing data for the problem statement:** To test and validate the database design, it is essential to have a representative dataset. We have prepared and collected at least 40 records for each table, simulating real-world data for the canteen management system. This data will be used to populate the tables and ensure that the database schema can handle various scenarios and edge cases effectively. We have gathered this data from the local canteen present within our college premises.
3. **Writing SQL queries and questions:** We have compiled a list of at least 100 questions that can be answered using SQL queries on the canteen management system database. These questions cover a wide range of scenarios and operations, including retrieving user and item information, managing orders and bills, querying staff and admin details, and performing various analytical tasks. By writing these queries, we have demonstrated the capability of the database schema to support diverse data retrieval and analysis requirements.

Throughout this process, we have created a database and populated the tables with the prepared data. This hands-on approach has allowed us to validate the database design, test the SQL queries, and ensure that the system can effectively handle the required operations and meet the specified requirements.

By normalizing the schema, preparing representative data, and writing SQL queries, we have achieved the following:

1. Established a solid foundation for the canteen management system database, ensuring data integrity, efficiency, and scalability.
2. Validated the database design by populating the tables with realistic data and testing various scenarios.
3. Demonstrated the capability of the database schema to support a wide range of queries and operations required for the canteen management system.

This comprehensive approach has provided us with a robust and well-designed database solution that can effectively manage and analyze data related to the canteen management system.

-----THE END-----