



**The Faculty of Information and Communication Technology
Mahidol University**

Project Assignment
Phase 2: Form Design to Implementation

6488021 Russarin Eaimrittikrai
6488045 Supithcha Jongphoemwatthanaphon
6488049 Kanita Karunkittikun
6488175 Penpitchapa Pantaraksakul

ITCS241 Database Management Systems
Dr. Preecha Tangworakitthaworn
Dr. Jidapa Kraisangka
December 2, 2022

Table of Contents

Abstract	2
Introduction to Bangkok Mass Transit System	3
The Revised ERD	5
8 Steps to transform ERD to relational schema	6
The Final Relation Schema	15
Data Dictionary	16
Table Description	20
SQL Queries	29
References	46

Link to the video presentation:

<https://drive.google.com/file/d/1qycf2H8RRt8EYTXelDBMgC5DaWDh7CuJ/view?usp=sharing>

Abstract

The presented report is a component of ITCS241 Database Management System to implement the real database to store data according to the information from the project phase one for the business process or application in the real world businesses. This report includes the Bangkok Mass Transit System business background, entity relationship diagram of BTS, 8 Steps to transform ERD to the final relational schema, Data Dictionary, Table Description, and SQL Queries.

Introduction to Bangkok Mass Transit System

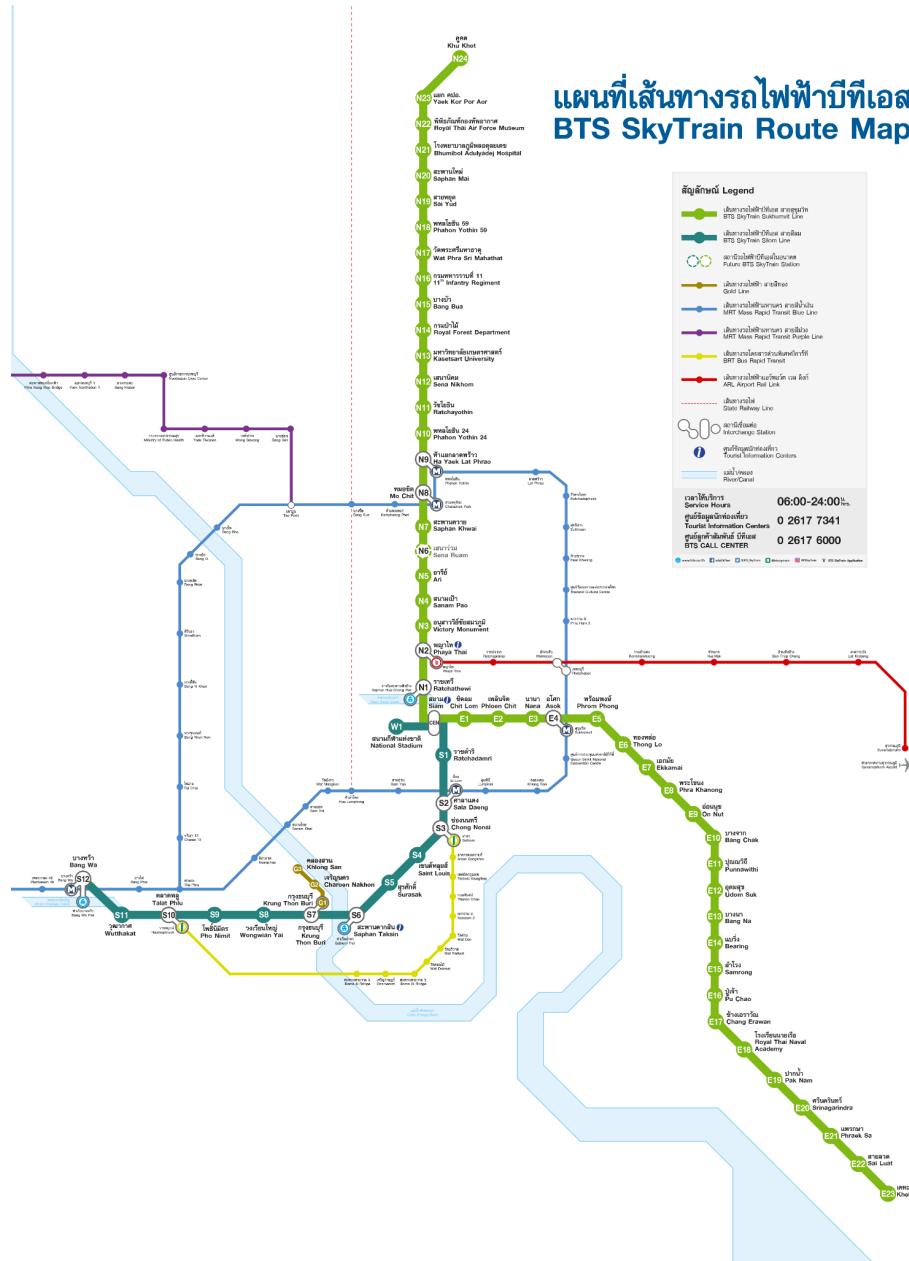


Bangkok Mass Transit System known as BTS Skytrain is the first uplifted rapid mass transit railway system in Bangkok, Thailand that Thai citizens have been using since 1999 until these days. It is managed by the Bangkok Mass Transit System Public Company Limited (BTSC).

In the beginning, the BTS Skytrain operated on 2 lines and 23 stations in total: 17 stations in Sukhumvit and 7 stations on the Silom line. However, it has been developed over time, 30 additional stations have been added on the Sukhumvit line, 6 additional stations have been added on the Silom line, and the latest 3 new stations on the Gold line.

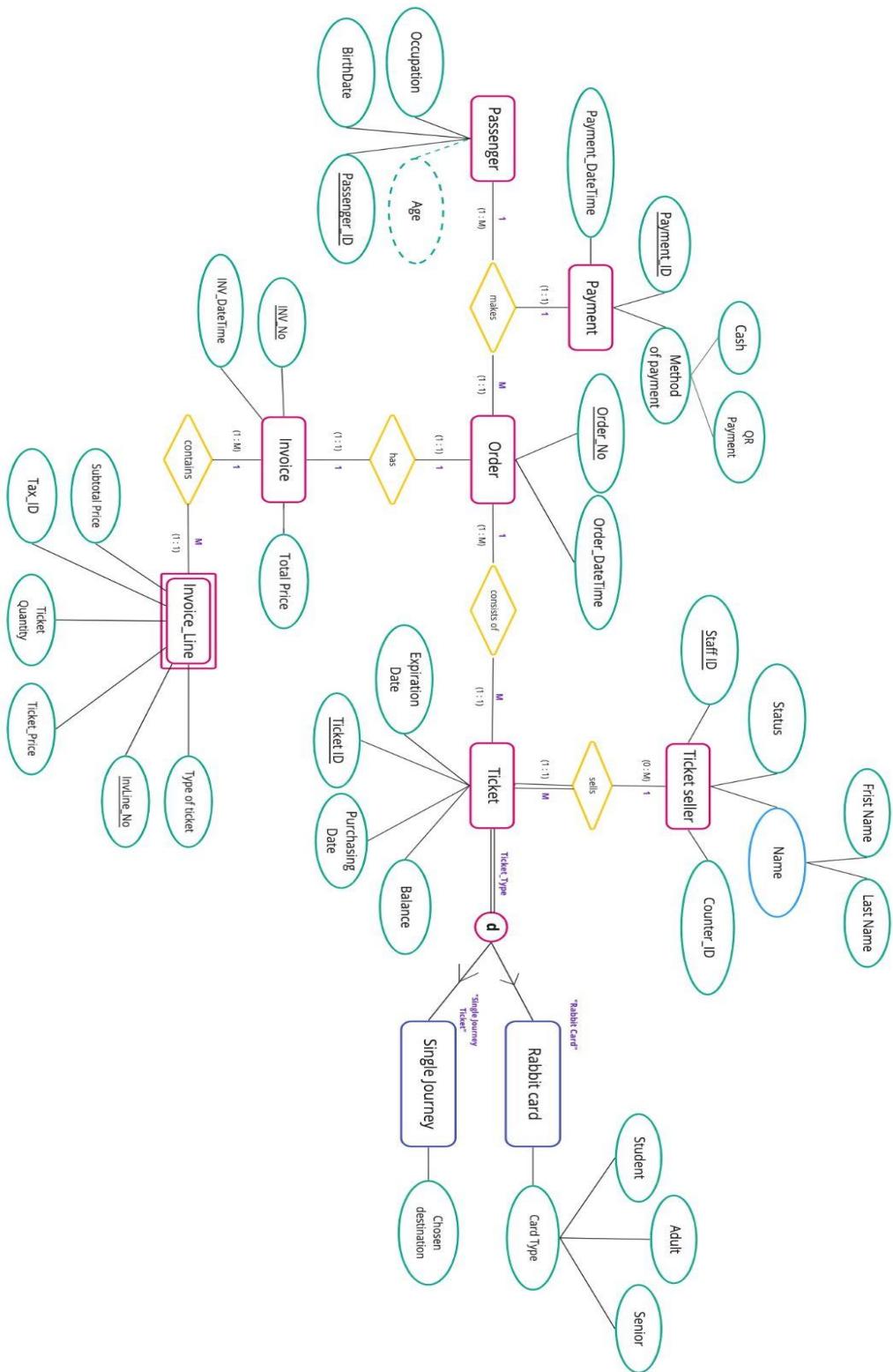
Passengers are allowed to purchase tickets at the BTS station by transforming funds into BTS SkyTrain tickets such as Rabbit Card or purchasing a one-time used ticket such as a Single Journey Ticket for passengers who want to travel between stations. The passengers are able to choose ways to purchase a ticket out of several options consisting of purchasing a ticket at the counter or using self-service at the station with cash, QR code scanning, or Rabbit card, and online.

แผนที่เส้นทางรถไฟฟ้าบีทีเอส BTS SkyTrain Route Map



The BTS SkyTrain Route Map

The Revised ERD



Link to ERD board: https://miro.com/app/board/uXjVPAzBFxk/?share_link_id=964655227048

8 Steps to transform ERD to relational schema

1. Transforming regular entity types

Passenger

Psg_ID	Psg_BD	Psg_Occupation
--------	--------	----------------

Payment

Pmt_ID	Method: cash	Method: QR payment	Pmt_DateTime
--------	--------------	--------------------	--------------

Order

Order_NO	Order_DateTime
----------	----------------

Invoice

INV_No	TotalPrice	INV_DateTime
--------	------------	--------------

Ticket

Ticket_ID	Purchasing_Date	Expiration_Date	Balance	Ticket_Type
-----------	-----------------	-----------------	---------	-------------

Ticket Seller

Staff_ID	Counter_ID	Tk_Status	FirstName	LastName
----------	------------	-----------	-----------	----------

2. Transforming weak entities

Passenger

Psg_ID	Psg_BD	Psg_Occupation
--------	--------	----------------

Payment

Pmt_ID	Method: cash	Method: QR payment	Pmt_DateTime
--------	--------------	--------------------	--------------

Order

Order_NO	Order_DateTime
----------	----------------

Invoice

INV_No	TotalPrice	INV_DateTime
--------	------------	--------------

Invoice Line

INVine_No	Ticket_Type	Ticket_Price	Tax_ID	SubTotal_price	Ticket_Quantity
-----------	-------------	--------------	--------	----------------	-----------------

<u>INV_No</u>

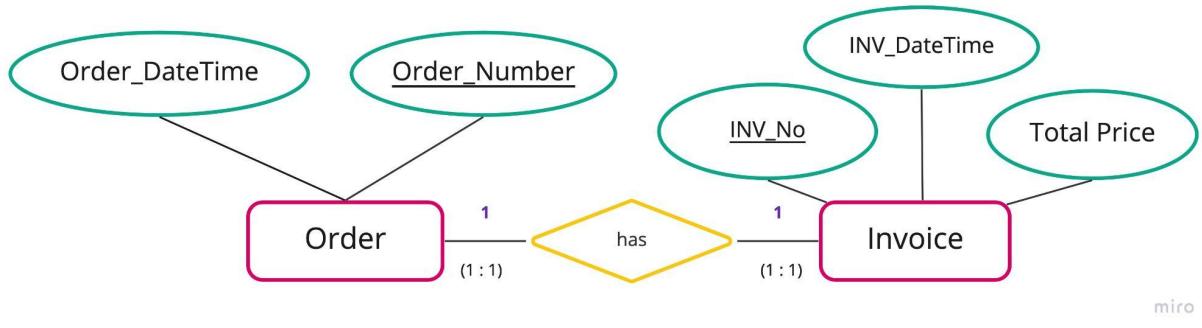
Ticket

Ticket_ID	Purchasing_Date	Expiration_Date	Balance	Ticket_Type
-----------	-----------------	-----------------	---------	-------------

Ticket Seller

Staff_ID	Counter_ID	Tk_Status	FirstName	LastName
----------	------------	-----------	-----------	----------

3. Transforming binary 1:1 relationship types



Passenger

Psg_ID	Psg_BD	Psg_Occupation
--------	--------	----------------

Payment

Pmt_ID	Method: cash	Method: QR payment	Pmt_DateTime
--------	--------------	--------------------	--------------

Order

Order_NO	Order_DateTime
----------	----------------

Invoice

INV_No	TotalPrice	INV_DateTime	Order_NO
--------	------------	--------------	----------

Invoice Line

INVLine_No	Ticket_Type	Ticket_Price	Tax_ID	SubTotal_price	Ticket_Quantity
------------	-------------	--------------	--------	----------------	-----------------

INV_No

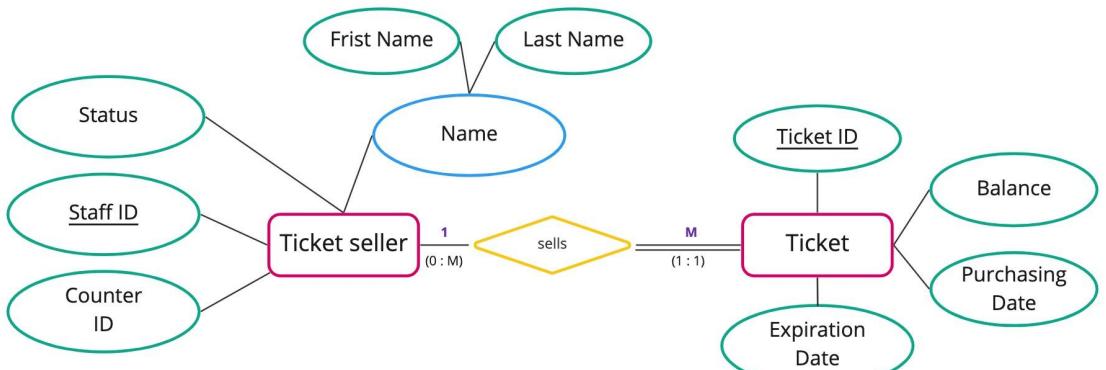
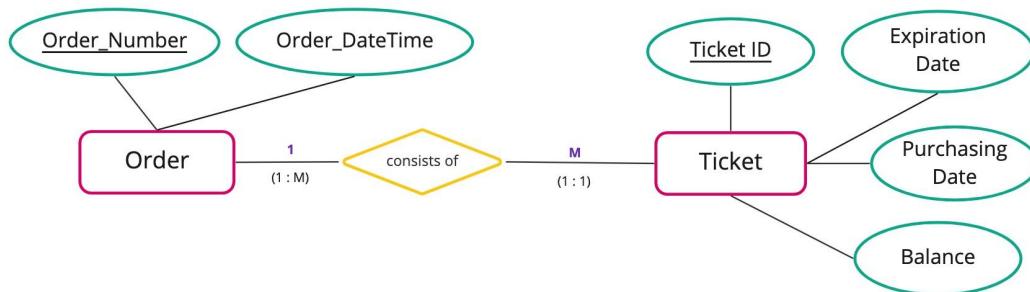
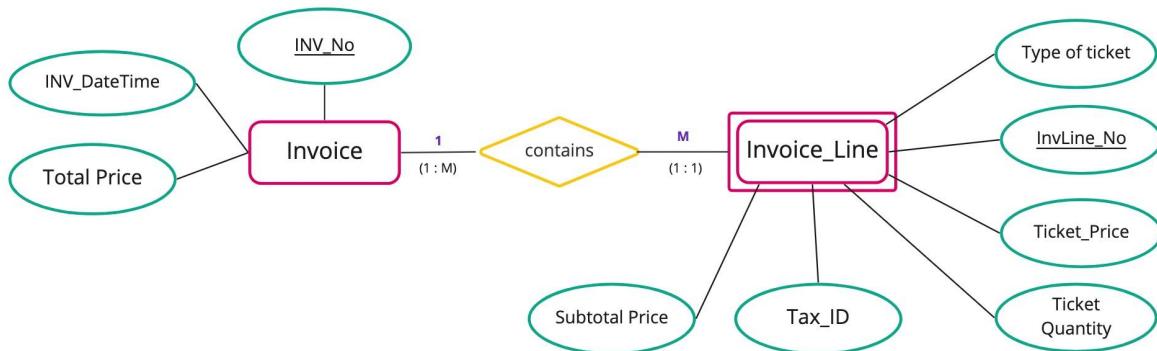
Ticket

Ticket_ID	Purchasing_Date	Expiration_Date	Balance	Ticket_Type
-----------	-----------------	-----------------	---------	-------------

Ticket Seller

Staff_ID	Counter_ID	Tk_Status	FirstName	LastName
----------	------------	-----------	-----------	----------

4. Transforming binary 1:N relationship types



miro

Passenger

<u>Psg_ID</u>	Psg_BD	Psg_Occupation
---------------	--------	----------------

Payment

<u>Pmt_ID</u>	Method: cash	Method: QR payment	Pmt_DateTime
---------------	--------------	--------------------	--------------

Order

<u>Order_NO</u>	Order_DateTime
-----------------	----------------

Invoice

<u>INV_No</u>	TotalPrice	INV_DateTime	<u>Order_NO</u>
---------------	------------	--------------	-----------------

Invoice Line

<u>INVLine_No</u>	Ticket_Type	Ticket_Price	Tax_ID	SubTotal_Price	Ticket_Quantity
-------------------	-------------	--------------	--------	----------------	-----------------

<u>INV_No</u>

Ticket

<u>Ticket_ID</u>	Purchasing_Date	Expiration_Date	Balance	Ticket_Type	<u>Staff_ID</u>	<u>Order_NO</u>
------------------	-----------------	-----------------	---------	-------------	-----------------	-----------------

Ticket Seller

<u>Staff_ID</u>	Counter_ID	Tk_Status	FirstName	LastName
-----------------	------------	-----------	-----------	----------

Ticket

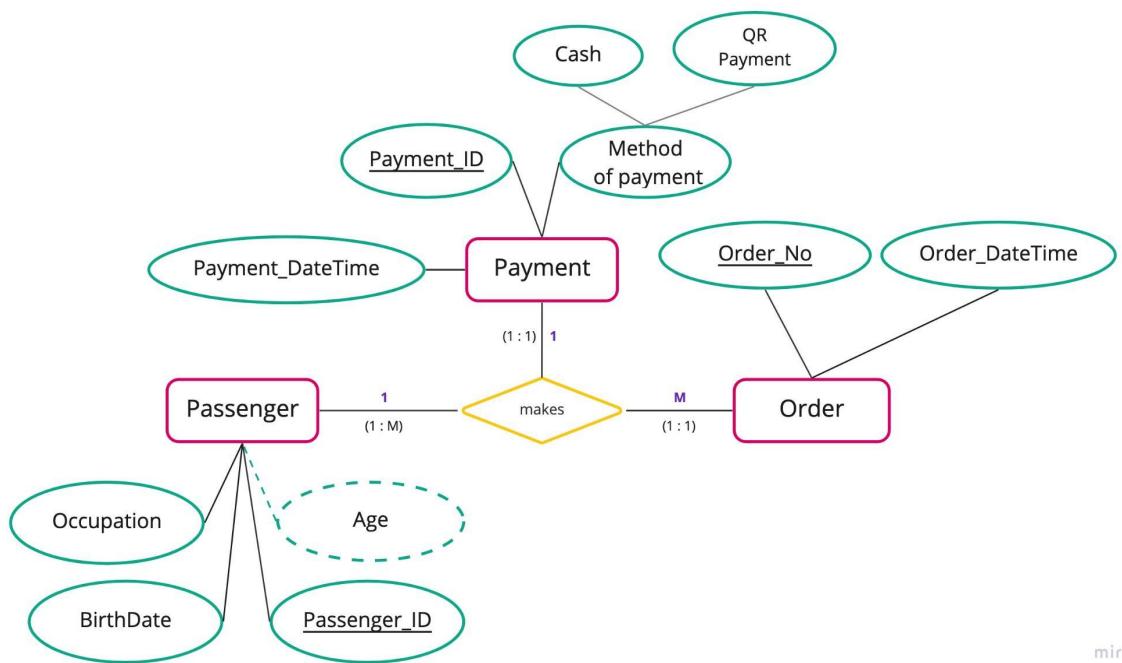
<u>Card_ID</u>	PurchasingDate	ExpirationDate	Price	Ticket_Type	<u>Order_NO</u>
----------------	----------------	----------------	-------	-------------	-----------------

Ticket Seller

<u>Staff_ID</u>	CounterID	Status	FirstName	LastName
-----------------	-----------	--------	-----------	----------

5. Transforming binary M:N relationship types (None)
6. Transforming multi-valued attribute A in relation K (None)

7. Transforming N-ary relationship types



miro

Passenger

<u>Psg_ID</u>	Psg_BD	Psg_Occupation
---------------	--------	----------------

Payment

<u>Pmt_ID</u>	Method: cash	Method: QR payment	Pmt_DateTime
---------------	--------------	--------------------	--------------

Order

<u>Order_NO</u>	Order_DateTime
-----------------	----------------

Makes

<u>Psg_ID</u>	<u>Pmt_ID</u>	<u>Order_NO</u>
---------------	---------------	-----------------

Invoice

<u>INV_No</u>	TotalPrice	INV_DateTime	<u>Order_NO</u>
---------------	------------	--------------	-----------------

Invoice Line

<u>INVine_No</u>	Ticket_Type	Ticket_Price	Tax_ID	SubTotal_price	Ticket_Quantity
------------------	-------------	--------------	--------	----------------	-----------------

<u>INV_No</u>

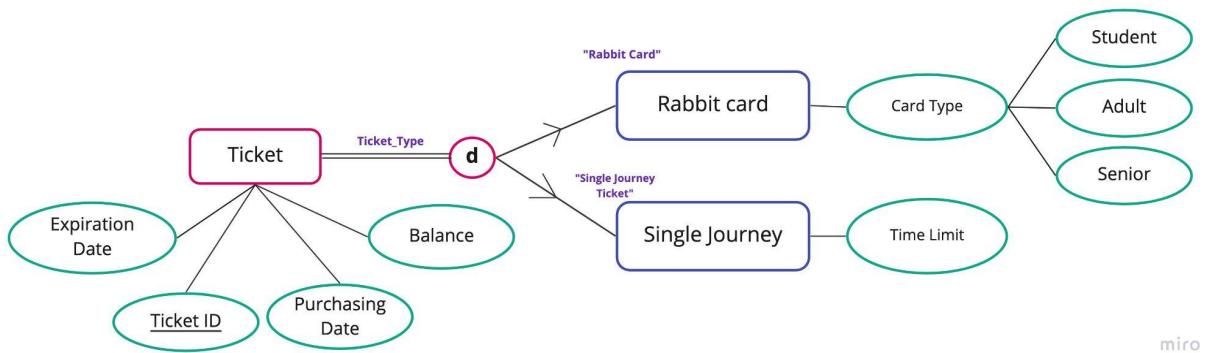
Ticket

<u>Ticket_ID</u>	Purchasing_Date	Expiration_Date	Balance	Ticket_Type	<u>Staff_ID</u>	<u>Order_NO</u>
------------------	-----------------	-----------------	---------	-------------	-----------------	-----------------

Ticket Seller

<u>Staff_ID</u>	Counter_ID	Tk_Status	FirstName	LastName
-----------------	------------	-----------	-----------	----------

8. Transforming Specialization or Generalization



Passenger

<u>Psg_ID</u>	Psg_BD	Psg_Occupation
---------------	--------	----------------

Payment

<u>Pmt_ID</u>	Method: cash	Method: QR payment	Pmt_DateTime
---------------	--------------	--------------------	--------------

Order

<u>Order_NO</u>	Order_DateTime
-----------------	----------------

Makes

<u>Psg_ID</u>	<u>Pmt_ID</u>	<u>Order_NO</u>
---------------	---------------	-----------------

Invoice

<u>INV_No</u>	TotalPrice	INV_DateTime	<u>Order_NO</u>
---------------	------------	--------------	-----------------

Invoice Line

<u>INVine_No</u>	Ticket_Type	Ticket_Price	Tax_ID	SubTotal_price	Ticket_Quantity
------------------	-------------	--------------	--------	----------------	-----------------

<u>INV_No</u>

Ticket

<u>Ticket_ID</u>	Purchasing_Date	Expiration_Date	Balance	Ticket_Type	<u>Staff_ID</u>	<u>Order_NO</u>
------------------	-----------------	-----------------	---------	-------------	-----------------	-----------------

Ticket Seller

<u>Staff_ID</u>	Counter_ID	Tk_Status	FirstName	LastName
-----------------	------------	-----------	-----------	----------

Rabbit Card

<u>Ticket_ID</u>	Card_Type: Student	Card_Type: Adult	Card_Type: Senior
------------------	--------------------	------------------	-------------------

Single Journey Ticket

<u>Ticket_ID</u>	Chosen destination
------------------	--------------------

The Final Relation Schema



Data Dictionary

Table Name	Attribute Name	Contents	Type	Format	Nulla ble	Range	Key	FK Reference Table
Passenger	Psg_ID	Passenger's ID	CHAR(9)	Xxxxxxx x			PK	
	Psg_BD	Passenger's birth date	DATE	Yyyy-mm- dd				
	Psg_Occupation	Passenger's occupation	VARCHAR (100)	Xxxxxxx	Y			
Payment	Pmt_ID	Payment's ID	CHAR(9)	Xxxxxxx x			PK	
	Pmt_Method	Method of payment	VARCHAR (50)	Xxxxxx				
	Pmt_DateTime	Date and time of payment	DATETIM E	Yyyy-mm- dd HH:MI:SS				
Setorder	Order_No	Order's number	VARCHAR (20)	Xxxxxx			PK	
	Order_DateTime	Order's date and time	DATETIM E	Yyyy-mm- dd HH:MI:SS				
Make	Psg_ID	Passenger's ID	CHAR(9)	Xxxxxxx x			FK	Psg_ID [Passenger]
	Pmt_ID	Payment's ID	CHAR(9)	Xxxxxxx x			FK	Pm_ID [Payment]
	Order_No	Number of Order	VARCHAR (20)	Xxxxxx			FK	Order_No [Order]
Invoice	Inv_NO	Number of Invoice	CHAR(20)	Xxx-x-xxx xxx-xxxx-x x			PK	
	TotalPrice	Total price in the invoice	DECIMAL(9,2)	1234567.8 9				

	Inv_DateTime	Date and time of invoice	DATETIME	YYYY-mm-dd HH:MI:SS				
	Order_No	Number of Order	VARCHAR(20)	Xxxxxxx		FK	Order_No [Order]	
Invoice Line	InvLine_No	Number of Invoice's line	CHAR(6)	Xxxxxxx		PK		
	Ticket_Type	Type of Ticket	VARCHAR(20)	Xxxxxxxxx				
	Ticket_Price	Price per ticket	DECIMAL(9,2)	1234567.89				
	Tax_ID	Tax ID in the Invoice	CHAR(13)	Xxxxxxxxxxxxx				
	Subtotal_price	Price of Subtotal	DECIMAL(9,2)	1234567.89				
	Ticket_Quantity	Quantity of ticket	INT	xx		1 to 10		
	Inv_NO	Number of Invoice	CHAR(20)	Xxx-x-xxx xxx-xxxx-x x		FK	Inv_NO [Invoice]	
Ticket	Ticket_ID	Ticket's ID	CHAR(12)	Xxxxxxxxxxxxx		PK		
	Purchasing_Date	Ticket's purchasing date	DATE	yyyy-mm-dd				
	Expiration_Date	Ticket's expiration date	DATE	yyyy-mm-dd				
	Balance	Ticket's balance	DECIMAL(10,2)	12345678.12				
	Ticket_Type	Type of ticket	VARCHAR(20)	Xxxxxxx		Rabbit Card,		

						Single Journey		
	Staff_ID	Ticket seller's staff ID	CHAR(6)	Xxxxxx	Y		FK	Staff_ID [Ticket Seller]
	Order_No	Number of Order	VARCHAR(20)	Xxxxxx			FK	Order_No [Order]
Ticket Seller	Staff_ID	Ticket seller's staff ID	CHAR(6)	Xxxxxx			PK	
	Counter_ID	Ticket seller's station counter ID	CHAR(3)	Xxxxxxxxxx x				
	Tk_Status	Ticket seller's status	CHAR(1)	x		O,C		
	FirstName	Ticket seller's first name	VARCHAR(100)	Xxxxxx				
	LastName	Ticket seller's last name	VARCHAR(100)	Xxxxxx				
Rabbit Card	Ticket_ID	Ticket's ID	CHAR(12)	Xxxxxxxxxx xxxx			FK	Ticket_ID [Ticket]
	Card_Type	Type of rabbit card	VARCHAR(10)	Xxxxxx		Student , Adult, Senior		
Single Journey Ticket	Ticket_ID	Ticket's ID	CHAR(12)	Xxxxxxxxxx xxxx			FK	Ticket_ID [Ticket]
	Destination	Chosen	VARCHAR	Xxxxxx				

		destination in the ticket	(20)						
--	--	------------------------------	------	--	--	--	--	--	--

Table Description

- Passenger Table

Passenger table is for collecting passengers' data including passenger ID which is the primary key, passengers' date of birth, and passengers' occupations.

```
10 • ⊖ CREATE TABLE passenger (
11     Psg_ID      CHAR(9)      NOT NULL ,
12     Psg_BD       DATE        NOT NULL ,
13     Psg_Occupation    VARCHAR(100) ,
14     CONSTRAINT Pk_Psg PRIMARY KEY (Psg_ID)
15 ) ;

112 •   INSERT INTO passenger (Psg_ID,  Psg_BD,  Psg_Occupation)
113     VALUES  ( "012345678", "2000-01-02", "Teacher" ) ,
114             ( "123456789", "1998-02-03", "Actor" ) ,
115             ( "234567891", "2002-03-04", "Student" ) ,
116             ( "345678912", "1995-04-05", "Pilot" ) ,
117             ( "456789123", "2004-05-06", "Student" ) ,
118             ( "567891234", "1997-06-07", "Singer" ) ,
119             ( "678912345", "2008-09-10", "Model" ) ,
120             ( "789123456", "1999-09-09", "Mason" ) ,
121             ( "891234567", "1999-10-20", NULL) ;
```

- Payment Table

Payment table is for collecting payment data including payment ID which is the primary key, method of payment, and Date and time of payment.

```
18 • ⊖ CREATE TABLE payment (
19     Pm_ID      CHAR(9)      NOT NULL ,
20     Pm_Method    VARCHAR(50) NOT NULL ,
21     Pm_DateTime   DATETIME    NOT NULL ,
22     CONSTRAINT Pk_Pm PRIMARY KEY (Pm_ID)
23 ) ;
```

```

158 •  INSERT INTO payment (Pm_ID, Pm_Method, Pm_DateTime)
159     VALUES  ( "012345678", "Cash", "2011-12-22 05:50:50" ) ,
160             ( "123456789", "Qr Code", "2011-11-23 06:55:59" ) ,
161             ( "234567891", "Cash", "2010-12-24 07:40:57" ) ,
162             ( "345678912", "Cash", "2010-11-25 08:44:47" ) ,
163             ( "456789123", "Qr Code", "2012-10-26 09:45:49" ) ,
164             ( "567891234", "Cash", "2012-11-27 10:35:45" ) ,
165             ( "678912345", "Cash", "2012-10-28 11:30:40" ) ,
166             ( "789123456", "Qr Code", "2013-09-29 12:33:43" ) ,
167             ( "891234567", "Qr Code", "2013-10-30 13:34:34" ) ,
168             ( "912345678", "Cash", "2014-09-30 14:35:56" ) ,

```

- Setorder Table

Setorder table is for collecting orders' data including the order number which is the primary key and the date and time that the order was created.

```

26 • CREATE TABLE setorder (
27     Order_No      CHAR(20)      NOT NULL ,
28     Order_DateTime DATETIME      NOT NULL ,
29     CONSTRAINT Pk_Order PRIMARY KEY (Order_No)
30 ) ;

```



```

125 •  INSERT INTO setorder (Order_No, Order_DateTime)
126     VALUES  ( "12345678", "2011-12-22 05:45:50" ),
127             ( "23456781", "2011-11-23 06:50:59" ),
128             ( "34567812", "2010-12-24 07:35:57" ),
129             ( "45678123", "2010-11-25 08:42:47" ),
130             ( "56781234", "2012-10-26 09:44:49" ),
131             ( "67812345", "2012-11-27 10:33:45" ),
132             ( "78123456", "2012-10-28 11:28:40" ),
133             ( "87654321", "2013-09-29 12:31:43" ),
134             ( "01234567", "2013-10-30 13:32:34" ),
135             ( "76543210", "2014-09-30 14:30:56" ),

```

- Make Table

Make table is for connecting primary keys from passenger table, payment table, and setorder table.

```

33 • CREATE TABLE make (
34     Psg_ID      CHAR(9)      NOT NULL ,
35     Pm_ID       CHAR(9)      NOT NULL ,
36     Order_No    CHAR(20)     NOT NULL ,
37     CONSTRAINT Fk_MakePsgID      FOREIGN KEY (Psg_ID)
38             REFERENCES passenger (Psg_ID) ,
39     CONSTRAINT Fk_MakePmID       FOREIGN KEY (Pm_ID)
40             REFERENCES payment   (Pm_ID) ,
41     CONSTRAINT Fk_MakeOrderNo   FOREIGN KEY (Order_No)
42             REFERENCES setorder (Order_No)
43 ) ;

```

```

332 • INSERT INTO make ( Psg_ID, Pm_ID, Order_No )
333     VALUES ( "012345678", "012345678", "12345678" ),
334             ( "123456789", "123456789", "23456781" ) ,
335             ( "234567891", "234567891", "34567812" ) ,
336             ( "345678912", "345678912", "45678123" ) ,
337             ( "456789123", "456789123", "56781234" ) ,
338             ( "567891234", "567891234", "67812345" ) ,
339             ( "678912345", "678912345", "78123456" ) ,
340             ( "789123456", "789123456", "87654321" ) ,
341             ( "891234567", "891234567", "01234567" ) ,
342             ( "912345678", "912345678", "76543210" ) ,

```

- Invoice Table

Invoice table is for collecting invoice data including the invoice number which is the primary key, the total price of the order, the date and time that the invoice has been created, and the order number which is the foreign key.

```

45 • CREATE TABLE invoice (
46     Inv_No          CHAR(20)      NOT NULL ,
47     TotalPrice     DECIMAL(9,2)   NOT NULL ,
48     Inv_Datetime   DATETIME     NOT NULL ,
49     Order_No       VARCHAR(20)   NOT NULL ,
50     CONSTRAINT Pk_Invoice      PRIMARY KEY (Inv_No),
51     CONSTRAINT Fk_InvoiceOrderID FOREIGN KEY (Order_No)
52         REFERENCES setorder    (Order_No)
53 ) ;
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191 • INSERT INTO invoice (Inv_No, TotalPrice, Inv_DateTime, Order_No)
192     VALUES ( "658-7-285612-7748-02", "665.00" , "2011-12-22 05:59:50" , "12345678"),
193             ( "898-5-984562-2348-42", "16.00" , "2011-11-23 06:57:59" , "23456781"),
194             ( "414-1-185123-2673-71", "637.00" , "2010-12-24 07:47:57" , "34567812"),
195             ( "263-4-485755-4648-16", "784.00" , "2010-11-25 08:46:47" , "45678123"),
196             ( "312-5-745212-6325-75", "189.00" , "2012-10-26 09:49:40" , "56781234"),
197             ( "498-2-324517-4421-46", "280.00" , "2012-11-27 10:38:00" , "67812345"),
198             ( "553-3-845302-3320-34", "400.00" , "2012-10-28 11:37:00" , "78123456"),
199             ( "854-6-984210-6849-89", "600.00" , "2013-09-29 12:36:33" , "87654321"),
200             ( "142-2-632105-1123-62", "920.00" , "2013-10-30 13:37:44" , "01234567"),
201             ( "651-7-756895-7584-25", "1059.00" , "2014-09-30 14:40:55" , "76543210"),
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
698
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
888
889
889
890
891
892
893
894
895
896
897
898
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2
```

- Invoice Line Table

InvoiceLine Table is for collecting invoice line data including Invoice line number which is the primary key, ticket type, ticket price, tax ID, subtotal price, ticket quantity, and invoice number which is the foreign key.

```

56 • CREATE TABLE invoiceline (
57     InvLine_No      CHAR(6)          NOT NULL ,
58     Ticket_Type    VARCHAR(20)       NOT NULL ,
59     Ticket_Price   DECIMAL(9,2)      NOT NULL ,
60     Tax_ID         CHAR(13)         NOT NULL ,
61     Subtotal_Price DECIMAL(9,2)      NOT NULL ,
62     Ticket_Quantity INT             NOT NULL ,
63     INV_No         CHAR(20)         NOT NULL ,
64     CONSTRAINT Pk_InvLine        PRIMARY KEY (InvLine_No),
65     CONSTRAINT Fk_invoiceINV_No FOREIGN KEY (INV_No)
66           REFERENCES invoice (INV_No)
67 );

```

```

225 • INSERT INTO invoiceline (InvLine_No, Ticket_Type, Ticket_Price, Tax_ID, Subtotal_Price, Ticket_Quantity, INV_No)
226 VALUES ( "012345", "Single Journey", "44.00", "0101112121122", "88.00", 2, "658-7-285612-7748-02" ) ,
227 ( "123456", "Single Journey", "59.00", "0101112121122", "177.00", 3, "658-7-285612-7748-02" ) ,
228 ( "234567", "Rabbit card", "200.00", "0101112121122", "400.00", 1, "658-7-285612-7748-02" ) ,
229 ( "345678", "Single Journey", "16.00", "0104442121122", "16.00", 1, "898-5-984562-2348-42" ) ,
230 ( "456780", "Rabbit card", "200.00", "0106662121152", "600.00", 3, "414-1-185123-2673-71" ) ,
231 ( "567890", "Single Journey", "37.00", "0106662121152", "37.00", 1, "414-1-185123-2673-71" ) ,
232 ( "678900", "Single Journey", "26.00", "0103332121129", "52.00", 2, "263-4-485755-4648-16" ) ,
233 ( "789012", "Rabbit card", "200.00", "0103332121129", "400.00", 2, "263-4-485755-4648-16" ) ,
234 ( "890123", "Rabbit card", "200.00", "0103332121129", "200.00", 1, "263-4-485755-4648-16" ) ,
235 ( "901234", "Single Journey", "33.00", "0103332121129", "132.00", 4, "263-4-485755-4648-16" ) ,

```

- Ticket Table

Ticket table is for collecting ticket data including the ticket ID which is the primary key, purchasing and expiration date of that ticket, the balance on the ticket, the ticket type, the order number which is the foreign key, and the staff ID who issued the ticket which is the foreign key.

- Ticket Seller Table

Ticket seller Table is for collecting the ticket sellers' data including staff ID which is the primary key, counter ID, ticket status which has 2 types: Open and Close, and first name and last name of the ticket sellers.

```

83 • CREATE TABLE ticketseller (
84     Staff_ID      CHAR(6)      NOT NULL ,
85     Counter_ID    CHAR(3)      NOT NULL ,
86     Tk_Status     CHAR(1)      NOT NULL ,
87     FirstName     VARCHAR(100)  NOT NULL ,
88     LastName      VARCHAR(100)  NOT NULL ,
89     CONSTRAINT Tk_Status CHECK (Tk_Status IN ('O','C')) ,
90     CONSTRAINT Pk_TkSellerStaffID          PRIMARY KEY (Staff_ID)
91 ) ;

```

```

319 • INSERT INTO ticketseller (Staff_ID, Counter_ID, Tk_Status, FirstName ,LastName)
320   VALUES  ( "134589", "N02", "0"," JESADA" , " NIMMANNITYA" ) ,
321           ( "645824", "E04", "0"," CHARAS" , "SUWANWELA" ) ,
322           ( "258246", "S10", "C"," PRADIST " , "SAKSRI" ) ,
323           ( "976582", "N13", "0"," THUMANOON" , "YONGCHAIYUDH" ) ,
324           ( "354876", "S12", "C"," ATT" , " KETUSINGHA" ) ,
325           ( "221034", "N07", "C"," CHARAS" , " SUWANWELA" ) ,
326           ( "756894", "E02", "0"," CHOTE" , " PARNICHKULI" ) ,
327           ( "110243", "N22", "0","VICHAI" , "VANADURONGWAN" ) ,
328           ( "452163", "S01", "0"," DILOK" , " PHUVANANDH" ) ,
329           ( "578421", "W01", "C","PREECHA" , " RUOKSPOLMUANG" );

```

- Rabbit Card Table

Rabbit card table is for collecting rabbit card data including ticket ID which is the foreign key and card type which are student, senior, and adult.

```
94 • CREATE TABLE rabbitcard (
95     Ticket_ID    CHAR(12)        NOT NULL ,
96     Card_Type    VARCHAR(10)      NOT NULL ,
97     CONSTRAINT Fk_RabbitCardTicket_ID      FOREIGN KEY (Ticket_ID)
98             REFERENCES ticket      (Ticket_ID)
99 );
|
```



```
365 •   INSERT INTO rabbitcard (Ticket_ID, Card_Type)
366     VALUES  ( "110000123456", "Adult" ) ,
367          ( "110000989876", "Student" ) ,
368          ( "110909898989", "Student" ) ,
369          ( "112223333344", "Adult" ) ,
370          ( "112345678934", "Adult" ) ,
371          ( "113430009898", "Senior" ) ,
372          ( "114565456543", "Student" ) ,
373          ( "117654321980", "Student" ) ,
374          ( "118765432190", "Student" ) ,
375          ( "119876543210", "Adult" );
```

- Single Journey Ticket Table

Single Journey Ticket Table is for collecting ticket' data including ticket ID which is the foreign key and the selected destination.

```

102 • CREATE TABLE singlejourneyticket (
103     Ticket_ID    CHAR(12)      NOT NULL ,
104     Destination  VARCHAR(20)   NOT NULL ,
105     CONSTRAINT Fk_SingleJourneyTicket_ID           FOREIGN KEY (Ticket_ID)
106             REFERENCES ticket    (Ticket_ID)
107 );
381 • INSERT INTO singlejourneyticket (Ticket_ID, Destination)
382     VALUES  ( "100012345678", "Ari"),
383          ( "100987654321", "Siam"),
384          ( "101000060004", "Chit Lom"),
385          ( "101011006696", "On Nut"),
386          ( "101011518624", "Bang na"),
387          ( "102000748553", "Krung Thon Buri"),
388          ( "102501511452", "Khu Khot"),
389          ( "107654321980", "Sala Daeng"),
390          ( "108765432190", "Bearing"),
391          ( "109876543210", "bang Wa");

```

SQL Queries

No.	Details	Code																																											
1	List of all passenger's ages.	<pre>SELECT CURDATE() AS Today,Psg_ID AS 'Passenger ID', Psg_BD AS 'Passenger's birthday', (YEAR(CURDATE()) - YEAR(Psg_BD)) AS Age FROM passenger ORDER BY (YEAR(CURDATE()) - YEAR(Psg_BD)) ASC ;</pre>																																											
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Today</th> <th>Passenger ID</th> <th>Passenger's Date of Birth</th> <th>Age</th> </tr> </thead> <tbody> <tr><td>2022-11-25</td><td>678912345</td><td>2008-09-10</td><td>14</td></tr> <tr><td>2022-11-25</td><td>456789123</td><td>2004-05-06</td><td>18</td></tr> <tr><td>2022-11-25</td><td>234567891</td><td>2002-03-04</td><td>20</td></tr> <tr><td>2022-11-25</td><td>012345678</td><td>2000-01-02</td><td>22</td></tr> <tr><td>2022-11-25</td><td>789123456</td><td>1999-09-09</td><td>23</td></tr> <tr><td>2022-11-25</td><td>891234567</td><td>1999-10-20</td><td>23</td></tr> <tr><td>2022-11-25</td><td>123456789</td><td>1998-02-03</td><td>24</td></tr> <tr><td>2022-11-25</td><td>567891234</td><td>1997-06-07</td><td>25</td></tr> <tr><td>2022-11-25</td><td>345678912</td><td>1995-04-05</td><td>27</td></tr> <tr><td>2022-11-25</td><td>912345678</td><td>1955-11-22</td><td>67</td></tr> </tbody> </table>		Today	Passenger ID	Passenger's Date of Birth	Age	2022-11-25	678912345	2008-09-10	14	2022-11-25	456789123	2004-05-06	18	2022-11-25	234567891	2002-03-04	20	2022-11-25	012345678	2000-01-02	22	2022-11-25	789123456	1999-09-09	23	2022-11-25	891234567	1999-10-20	23	2022-11-25	123456789	1998-02-03	24	2022-11-25	567891234	1997-06-07	25	2022-11-25	345678912	1995-04-05	27	2022-11-25	912345678	1955-11-22	67
Today	Passenger ID	Passenger's Date of Birth	Age																																										
2022-11-25	678912345	2008-09-10	14																																										
2022-11-25	456789123	2004-05-06	18																																										
2022-11-25	234567891	2002-03-04	20																																										
2022-11-25	012345678	2000-01-02	22																																										
2022-11-25	789123456	1999-09-09	23																																										
2022-11-25	891234567	1999-10-20	23																																										
2022-11-25	123456789	1998-02-03	24																																										
2022-11-25	567891234	1997-06-07	25																																										
2022-11-25	345678912	1995-04-05	27																																										
2022-11-25	912345678	1955-11-22	67																																										
2	List of the passenger ID of passengers who can purchase a student rabbit card. (Persons eligible to use the Rabbit Card For students are students under the age of 23).	<pre>SELECT Psg_ID AS 'Passenger's ID', (YEAR(CURDATE()) - YEAR(Psg_BD)) AS 'Age under 23' FROM passenger WHERE (YEAR(CURDATE()) - YEAR(Psg_BD)) < 23 ORDER BY (YEAR(CURDATE()) - YEAR(Psg_BD)) ASC ;</pre>																																											
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Passenger's ID</th> <th>Age under 23</th> </tr> </thead> <tbody> <tr><td>678912345</td><td>14</td></tr> <tr><td>456789123</td><td>18</td></tr> <tr><td>234567891</td><td>20</td></tr> <tr><td>012345678</td><td>22</td></tr> </tbody> </table>		Passenger's ID	Age under 23	678912345	14	456789123	18	234567891	20	012345678	22																																		
Passenger's ID	Age under 23																																												
678912345	14																																												
456789123	18																																												
234567891	20																																												
012345678	22																																												
3	List of orders that were made in the year between 2020 to 2022.	<pre>SELECT Order_No, YEAR(Order_DateTime) AS 'Year', DATE(Order_DateTime) AS 'Date', TIME(Order_DateTime) AS 'Time' FROM setorder WHERE YEAR(Order_DateTime) >= 2020 AND YEAR(Order_DateTime) <= 2022 ORDER BY YEAR(Order_DateTime) ASC ;</pre>																																											

Order_No	Year	Date	Time
01476421	2020	2020-03-10	19:10:16
31542012	2020	2020-04-09	20:12:17
48712052	2020	2020-04-11	18:09:15
87064259	2020	2020-05-08	21:10:17
66795214	2021	2021-12-16	11:11:06
98621052	2021	2021-03-12	17:00:14
34681205	2022	2022-10-16	11:00:07
91420512	2022	2022-11-15	12:00:08

4	List List of the ticket IDs of the Rabbit card that have the balance on their ticket. (Balance > 0)	<pre>SELECT Ticket_ID AS 'Ticket ID', balance AS Balance FROM ticket WHERE balance > 0 ORDER BY balance ASC ;</pre>																								
		<table border="1"> <thead> <tr> <th>Ticket ID</th> <th>Balance</th> </tr> </thead> <tbody> <tr> <td>118765432190</td> <td>23.50</td> </tr> <tr> <td>112223333344</td> <td>100.00</td> </tr> <tr> <td>112345678934</td> <td>250.14</td> </tr> <tr> <td>119876543210</td> <td>300.00</td> </tr> <tr> <td>117654321980</td> <td>342.21</td> </tr> <tr> <td>101011518624</td> <td>345.11</td> </tr> <tr> <td>110000123456</td> <td>490.80</td> </tr> <tr> <td>110909898989</td> <td>605.00</td> </tr> <tr> <td>113430009898</td> <td>1150.43</td> </tr> <tr> <td>110000989876</td> <td>1420.00</td> </tr> <tr> <td>114565456543</td> <td>2250.12</td> </tr> </tbody> </table>	Ticket ID	Balance	118765432190	23.50	112223333344	100.00	112345678934	250.14	119876543210	300.00	117654321980	342.21	101011518624	345.11	110000123456	490.80	110909898989	605.00	113430009898	1150.43	110000989876	1420.00	114565456543	2250.12
Ticket ID	Balance																									
118765432190	23.50																									
112223333344	100.00																									
112345678934	250.14																									
119876543210	300.00																									
117654321980	342.21																									
101011518624	345.11																									
110000123456	490.80																									
110909898989	605.00																									
113430009898	1150.43																									
110000989876	1420.00																									
114565456543	2250.12																									
5	List of the ticket IDs that have an expiration year before 2022.	<pre>SELECT Ticket_ID AS 'Ticket ID', YEAR(Expiration_Date) AS 'Year',Expiration_Date AS 'Expiration Date' FROM ticket WHERE YEAR(Expiration_Date) < 2022 ORDER BY YEAR(Expiration_Date) DESC;</pre>																								

Ticket ID	Year	Expiration Date
113430009898	2021	2021-09-30
100987654321	2020	2020-05-08
112223333344	2020	2020-08-31
100012345678	2019	2019-06-07
110000989876	2019	2019-10-26
114565456543	2019	2019-10-28
110909898989	2018	2018-11-23
112345678934	2018	2018-12-22
107654321980	2017	2017-06-05
108765432190	2016	2016-08-03
109876543210	2015	2015-07-22
101011518624	2013	2013-10-30
102501511452	2013	2013-09-29
101011006696	2012	2012-11-27
101000060004	2010	2010-11-25
102000748553	2010	2010-12-24

6	List of the orders that have the ticket quantity equals to 1.	<pre>SELECT Order_No AS 'Order No', Ticket_Quantity AS 'Ticket Quantity', Date(Inv_Datetime) AS 'Date' FROM invoice inv INNER JOIN inviceline invl ON inv.INV_No = invl.INV_No WHERE Ticket_Quantity = 1 ;</pre>
---	---	--

Order No	Ticket Quantity	Date
56781234	1	2012-10-26
67812345	1	2012-11-27
12345678	1	2011-12-22
21035412	1	2015-09-01
68597521	1	2015-07-22
98745031	1	2016-07-04
23456781	1	2011-11-23
71025406	1	2019-06-07
48712052	1	2020-04-11
34567812	1	2010-12-24
36502417	1	2019-02-14
34681205	1	2022-10-16
45678123	1	2010-11-25

7	List of the orders of passengers that purchased Rabbit cards.	<pre>SELECT o.Order_No AS 'Order No', m.Psg_ID AS 'Passenger ID', t.Ticket_Type AS 'Ticket type' FROM setorder o INNER JOIN make m ON m.Order_No = o.Order_No INNER JOIN ticket t ON t.Order_No = o.Order_No WHERE Ticket_Type = "Rabbit card"</pre>
---	---	--

		ORDER BY o.Order_No ASC ;																														
		<table border="1"> <thead> <tr> <th>Order No</th> <th>Passenger ID</th> <th>Ticket type</th> </tr> </thead> <tbody> <tr><td>12345678</td><td>012345678</td><td>Rabbit card</td></tr> <tr><td>23456781</td><td>123456789</td><td>Rabbit card</td></tr> <tr><td>56781234</td><td>456789123</td><td>Rabbit card</td></tr> <tr><td>65412743</td><td>678912345</td><td>Rabbit card</td></tr> <tr><td>68597521</td><td>234567891</td><td>Rabbit card</td></tr> <tr><td>76543210</td><td>912345678</td><td>Rabbit card</td></tr> <tr><td>78123456</td><td>678912345</td><td>Rabbit card</td></tr> <tr><td>91420512</td><td>789123456</td><td>Rabbit card</td></tr> <tr><td>98765423</td><td>567891234</td><td>Rabbit card</td></tr> </tbody> </table>	Order No	Passenger ID	Ticket type	12345678	012345678	Rabbit card	23456781	123456789	Rabbit card	56781234	456789123	Rabbit card	65412743	678912345	Rabbit card	68597521	234567891	Rabbit card	76543210	912345678	Rabbit card	78123456	678912345	Rabbit card	91420512	789123456	Rabbit card	98765423	567891234	Rabbit card
Order No	Passenger ID	Ticket type																														
12345678	012345678	Rabbit card																														
23456781	123456789	Rabbit card																														
56781234	456789123	Rabbit card																														
65412743	678912345	Rabbit card																														
68597521	234567891	Rabbit card																														
76543210	912345678	Rabbit card																														
78123456	678912345	Rabbit card																														
91420512	789123456	Rabbit card																														
98765423	567891234	Rabbit card																														
8	The total number of orders that passengers purchased was the Rabbit card and Single journey ticket.	<pre>SELECT COUNT(Order_No) AS 'Number of order', Ticket_Type AS 'Ticket type' FROM invoice inv INNER JOIN invoiceinvl ON inv.INV_No = invl.INV_No GROUP BY Ticket_Type ORDER BY COUNT(Order_No) ASC ;</pre>																														
		<table border="1"> <thead> <tr> <th>Number of order</th> <th>Ticket type</th> </tr> </thead> <tbody> <tr><td>24</td><td>Rabbit card</td></tr> <tr><td>42</td><td>Single Journey</td></tr> </tbody> </table>	Number of order	Ticket type	24	Rabbit card	42	Single Journey																								
Number of order	Ticket type																															
24	Rabbit card																															
42	Single Journey																															
9	List of the order IDs that have the highest total Price in each year.	<pre>SELECT YEAR(o.Order_DateTime) AS Year, o.Order_No AS 'Order ID', MAX(TotalPrice) AS 'The highest total price' FROM setorder o INNER JOIN invoice inv ON inv.Order_No = o.Order_No GROUP BY YEAR(o.Order_DateTime) ORDER BY YEAR(o.Order_DateTime) ASC ;</pre>																														

Year	Order ID	The highest total price
2010	34567812	784.00
2011	12345678	665.00
2012	56781234	400.00
2013	01234567	920.00
2014	76543210	1059.00
2015	21035412	1244.00
2016	32041574	1000.00
2017	98765423	400.00
2018	14702150	400.00
2019	36502417	1000.00
2020	01476421	1246.00
2021	66795214	548.00
2022	34681205	1435.00

10	The date and amount of cases that the passenger went to the destination "Siam".	<pre>SELECT DATE(o.Order_DateTime) AS `Date`, COUNT(t.Order_No) AS `Amount` ,Destination FROM ticket t INNER JOIN setorder o ON o.Order_No = t.Order_No INNER JOIN singlejourneyticket s ON s.Ticket_ID = t.Ticket_ID WHERE s.Destination = "Siam" GROUP BY DATE(o.Order_DateTime);</pre>
----	---	---

Date	Amount	Destination
2016-07-04	1	Siam

11	List of the ticket sellers (Staff ID, Counter ID, First Name, and Last Name) that are open.	<pre>SELECT Staff_ID AS "Staff ID", Counter_ID AS "Counter ID", CONCAT(FirstName, " ", LastName) AS "Full Name" FROM ticketseller WHERE Tk_Status = "O";</pre>
----	---	--

Staff ID	Counter ID	Full Name
110243	N22	VICHAI VANADURONGWAN
134589	N02	JESADA NIMMANNITYA
452163	S01	DILOK PHUVANANDH
645824	E04	CHARAS SUWANWELA
756894	E02	CHOTE PARNICKULI
976582	N13	THUMANOON YONGCHAIYUDH

12	The numbers of Cash payments and QR Code payments that were made before 2020.	<pre>SELECT Pm_Method AS "Payment Method", COUNT(Pm_ID) AS "Total of payment before 2020" FROM payment WHERE YEAR(Pm_DateTime) < "2020" GROUP BY Pm_Method;</pre>																																												
		<table> <thead> <tr> <th>Payment Method</th><th>Total of payment before 2020</th></tr> </thead> <tbody> <tr> <td>Cash</td><td>13</td></tr> <tr> <td>Qr Code</td><td>9</td></tr> </tbody> </table>	Payment Method	Total of payment before 2020	Cash	13	Qr Code	9																																						
Payment Method	Total of payment before 2020																																													
Cash	13																																													
Qr Code	9																																													
13	List of the top 10 invoices that have the highest value of the total price.	<pre>SELECT Inv_No AS "Invoice ID", TotalPrice AS "Total Price", Inv_DateTime AS "Invoice Date-Time", Order_No AS "Order Number" FROM invoice ORDER BY TotalPrice DESC LIMIT 10;</pre>																																												
		<table> <thead> <tr> <th>Invoice ID</th><th>Total Price</th><th>Invoice Date-Time</th><th>Order Number</th></tr> </thead> <tbody> <tr><td>236-9-510564-7789-56</td><td>1435.00</td><td>2022-11-15 12:08:00</td><td>91420512</td></tr> <tr><td>177-9-567890-2348-56</td><td>1246.00</td><td>2020-05-08 21:28:33</td><td>87064259</td></tr> <tr><td>456-4-567890-1234-56</td><td>1244.00</td><td>2015-09-01 16:22:34</td><td>21035412</td></tr> <tr><td>651-7-756895-7584-25</td><td>1059.00</td><td>2014-09-30 14:40:55</td><td>76543210</td></tr> <tr><td>122-7-324517-1234-71</td><td>1000.00</td><td>2016-08-03 18:22:11</td><td>32041574</td></tr> <tr><td>669-9-313326-4452-56</td><td>1000.00</td><td>2019-12-14 13:12:00</td><td>65412743</td></tr> <tr><td>142-2-632105-1123-62</td><td>920.00</td><td>2013-10-30 13:37:44</td><td>01234567</td></tr> <tr><td>166-9-485755-1234-56</td><td>837.00</td><td>2019-06-07 22:21:44</td><td>71025406</td></tr> <tr><td>263-4-485755-4648-16</td><td>784.00</td><td>2010-11-25 08:46:47</td><td>45678123</td></tr> <tr><td>199-5-324517-1234-56</td><td>748.00</td><td>2020-03-10 19:19:55</td><td>01476421</td></tr> </tbody> </table>	Invoice ID	Total Price	Invoice Date-Time	Order Number	236-9-510564-7789-56	1435.00	2022-11-15 12:08:00	91420512	177-9-567890-2348-56	1246.00	2020-05-08 21:28:33	87064259	456-4-567890-1234-56	1244.00	2015-09-01 16:22:34	21035412	651-7-756895-7584-25	1059.00	2014-09-30 14:40:55	76543210	122-7-324517-1234-71	1000.00	2016-08-03 18:22:11	32041574	669-9-313326-4452-56	1000.00	2019-12-14 13:12:00	65412743	142-2-632105-1123-62	920.00	2013-10-30 13:37:44	01234567	166-9-485755-1234-56	837.00	2019-06-07 22:21:44	71025406	263-4-485755-4648-16	784.00	2010-11-25 08:46:47	45678123	199-5-324517-1234-56	748.00	2020-03-10 19:19:55	01476421
Invoice ID	Total Price	Invoice Date-Time	Order Number																																											
236-9-510564-7789-56	1435.00	2022-11-15 12:08:00	91420512																																											
177-9-567890-2348-56	1246.00	2020-05-08 21:28:33	87064259																																											
456-4-567890-1234-56	1244.00	2015-09-01 16:22:34	21035412																																											
651-7-756895-7584-25	1059.00	2014-09-30 14:40:55	76543210																																											
122-7-324517-1234-71	1000.00	2016-08-03 18:22:11	32041574																																											
669-9-313326-4452-56	1000.00	2019-12-14 13:12:00	65412743																																											
142-2-632105-1123-62	920.00	2013-10-30 13:37:44	01234567																																											
166-9-485755-1234-56	837.00	2019-06-07 22:21:44	71025406																																											
263-4-485755-4648-16	784.00	2010-11-25 08:46:47	45678123																																											
199-5-324517-1234-56	748.00	2020-03-10 19:19:55	01476421																																											
14	The average price that people paid for tickets in 2021.	<pre>SELECT YEAR(Inv_DateTime) AS "Year", ROUND(AVG(TotalPrice), 2) AS "Average price" FROM invoice WHERE YEAR(Inv_DateTime) = "2021";</pre>																																												
		<table> <thead> <tr> <th>Year</th><th>Average price</th></tr> </thead> <tbody> <tr><td>2021</td><td>472.00</td></tr> </tbody> </table>	Year	Average price	2021	472.00																																								
Year	Average price																																													
2021	472.00																																													
15	The total number of orders that were made in 2018 to 2022.	<pre>SELECT YEAR(Order_DateTime) AS "Year", COUNT(Order_No) AS "Total number of order" FROM setorder WHERE YEAR(Order_DateTime) <= "2022" AND YEAR(Order_DateTime) >= "2018"</pre>																																												

		GROUP BY YEAR(Order_DateTime) ORDER BY YEAR(Order_DateTime) ASC;																		
		<table border="1"> <thead> <tr> <th>Year</th><th>Total number of order</th></tr> </thead> <tbody> <tr> <td>2018</td><td>3</td></tr> <tr> <td>2019</td><td>3</td></tr> <tr> <td>2020</td><td>4</td></tr> <tr> <td>2021</td><td>2</td></tr> <tr> <td>2022</td><td>2</td></tr> </tbody> </table>	Year	Total number of order	2018	3	2019	3	2020	4	2021	2	2022	2						
Year	Total number of order																			
2018	3																			
2019	3																			
2020	4																			
2021	2																			
2022	2																			
16	List of the passengers (Passenger ID, Rabbit card ID, Balance) who have the Student Rabbit Card.	<pre>SELECT Psg_ID AS "Passenger ID", t.Ticket_ID AS "Student Ticket ID", Balance FROM make m LEFT OUTER JOIN ticket t ON m.Order_No = t.Order_No INNER JOIN rabbitcard r ON t.Ticket_ID = r.Ticket_ID WHERE Card_Type = "Student";</pre>																		
		<table border="1"> <thead> <tr> <th>Passenger ID</th><th>Student Ticket ID</th><th>Balance</th></tr> </thead> <tbody> <tr> <td>456789123</td><td>110000989876</td><td>1420.00</td></tr> <tr> <td>123456789</td><td>110909898989</td><td>605.00</td></tr> <tr> <td>678912345</td><td>114565456543</td><td>2250.12</td></tr> <tr> <td>234567891</td><td>117654321980</td><td>342.21</td></tr> <tr> <td>012345678</td><td>118765432190</td><td>23.50</td></tr> </tbody> </table>	Passenger ID	Student Ticket ID	Balance	456789123	110000989876	1420.00	123456789	110909898989	605.00	678912345	114565456543	2250.12	234567891	117654321980	342.21	012345678	118765432190	23.50
Passenger ID	Student Ticket ID	Balance																		
456789123	110000989876	1420.00																		
123456789	110909898989	605.00																		
678912345	114565456543	2250.12																		
234567891	117654321980	342.21																		
012345678	118765432190	23.50																		
17	List of the passengers (Passenger ID, Rabbit card ID, Balance) who have the Adult Rabbit Card.	<pre>SELECT Psg_ID AS "Passenger ID", t.Ticket_ID AS "Adult Ticket ID", Balance FROM make m LEFT OUTER JOIN ticket t ON m.Order_No = t.Order_No INNER JOIN rabbitcard r ON t.Ticket_ID = r.Ticket_ID WHERE Card_Type = "Adult";</pre>																		
		<table border="1"> <thead> <tr> <th>Passenger ID</th><th>Adult Ticket ID</th><th>Balance</th></tr> </thead> <tbody> <tr> <td>567891234</td><td>110000123456</td><td>490.80</td></tr> <tr> <td>678912345</td><td>112223333344</td><td>100.00</td></tr> <tr> <td>012345678</td><td>112345678934</td><td>250.14</td></tr> <tr> <td>789123456</td><td>119876543210</td><td>300.00</td></tr> </tbody> </table>	Passenger ID	Adult Ticket ID	Balance	567891234	110000123456	490.80	678912345	112223333344	100.00	012345678	112345678934	250.14	789123456	119876543210	300.00			
Passenger ID	Adult Ticket ID	Balance																		
567891234	110000123456	490.80																		
678912345	112223333344	100.00																		
012345678	112345678934	250.14																		
789123456	119876543210	300.00																		
18	List of the passengers (Passenger ID, Rabbit card ID, Balance) who have the Senior Rabbit Card.	<pre>SELECT Psg_ID AS "Passenger ID", t.Ticket_ID AS "Senior Ticket ID", Balance FROM make m</pre>																		

		<pre>LEFT OUTER JOIN ticket t ON m.Order_No = t.Order_No INNER JOIN rabbitcard r ON t.Ticket_ID = r.Ticket_ID WHERE Card_Type = "Senior";</pre>																		
		<table border="1"> <thead> <tr> <th>Passenger ID</th> <th>Senior Ticket ID</th> <th>Balance</th> </tr> </thead> <tbody> <tr> <td>912345678</td> <td>113430009898</td> <td>1150.43</td> </tr> </tbody> </table>	Passenger ID	Senior Ticket ID	Balance	912345678	113430009898	1150.43												
Passenger ID	Senior Ticket ID	Balance																		
912345678	113430009898	1150.43																		
19	List of the top 5 order numbers that bought the highest quantity of Single Journey Tickets.	<pre>SELECT m.Order_No AS "Order ID", COUNT(Ticket_Quantity) AS "Ticket Quantity", SUM(Ticket_Price*Ticket_Quantity) AS "Ticket Price" FROM make m LEFT OUTER JOIN invoice i ON m.Order_No = i.Order_No LEFT OUTER JOIN invoiceLine il ON i.INV_No = il.INV_No WHERE Ticket_Type = "Single Journey" GROUP BY m.Order_No ORDER BY COUNT(Ticket_Quantity) DESC LIMIT 5;</pre>																		
		<table border="1"> <thead> <tr> <th>Order ID</th> <th>Ticket Quantity</th> <th>Ticket Price</th> </tr> </thead> <tbody> <tr> <td>91420512</td> <td>4</td> <td>835.00</td> </tr> <tr> <td>56781234</td> <td>3</td> <td>189.00</td> </tr> <tr> <td>87064259</td> <td>3</td> <td>646.00</td> </tr> <tr> <td>14702150</td> <td>3</td> <td>223.00</td> </tr> <tr> <td>12345678</td> <td>2</td> <td>265.00</td> </tr> </tbody> </table>	Order ID	Ticket Quantity	Ticket Price	91420512	4	835.00	56781234	3	189.00	87064259	3	646.00	14702150	3	223.00	12345678	2	265.00
Order ID	Ticket Quantity	Ticket Price																		
91420512	4	835.00																		
56781234	3	189.00																		
87064259	3	646.00																		
14702150	3	223.00																		
12345678	2	265.00																		
20	List of the total of the orders that were made and the total money paid in each year.	<pre>SELECT YEAR(Order_DateTime) AS "Year", COUNT(i.Order_No) AS "Total number of Order", SUM(TotalPrice) AS "Total number of income from orders" FROM setorder s LEFT OUTER JOIN invoice i ON s.Order_No = i.Order_No GROUP BY YEAR(Order_DateTime) ORDER BY YEAR(Order_DateTime) ASC;</pre>																		

Year	Total number of Order	Total number of income from orders
2010	2	1421.00
2011	2	681.00
2012	3	869.00
2013	2	1520.00
2014	2	1641.00
2015	2	1488.00
2016	2	1332.00
2017	1	400.00
2018	3	798.00
2019	3	2239.00
2020	4	2595.00
2021	2	944.00
2022	2	1747.00

21	List of the passenger ID of passengers who can purchase a senior rabbit card. (Anyone whose age is greater than or equal to (\geq) 60 will eligible to use the Rabbit Card for Senior)	<pre>SELECT Psg_ID AS 'Passenger's ID', (YEAR(CURDATE()) - YEAR(Psg_BD)) AS 'Age more than 60' FROM passenger WHERE (YEAR(CURDATE()) - YEAR(Psg_BD)) >= 60 ORDER BY (YEAR(CURDATE()) - YEAR(Psg_BD)) ASC ;</pre> <table border="1"> <thead> <tr> <th>Passenger's ID</th><th>Age more than 60</th></tr> </thead> <tbody> <tr> <td>912345678</td><td>67</td></tr> </tbody> </table>	Passenger's ID	Age more than 60	912345678	67		
Passenger's ID	Age more than 60							
912345678	67							
22	List of the passengers who are the students.	<pre>SELECT Psg_ID AS 'Passenger ID', Psg_Occupation AS 'Passenger Occupation' FROM passenger WHERE Psg_Occupation = "Student" ORDER BY Psg_ID ASC ;</pre> <table border="1"> <thead> <tr> <th>Passenger ID</th><th>Passenger Occupation</th></tr> </thead> <tbody> <tr> <td>234567891</td><td>Student</td></tr> <tr> <td>456789123</td><td>Student</td></tr> </tbody> </table>	Passenger ID	Passenger Occupation	234567891	Student	456789123	Student
Passenger ID	Passenger Occupation							
234567891	Student							
456789123	Student							
23	List of all the ticket sellers who sold tickets.	<pre>SELECT DISTINCT t.Staff_ID AS "Staff ID", CONCAT(FirstName, " ", LastName) AS "Full Name" FROM ticket t INNER JOIN ticketseller ts ON t.Staff_ID = ts.Staff_ID;</pre>						

Staff ID	Full Name
110243	VICHAI VANADURONGWAN
134589	JESADA NIMMANNITYA
221034	CHARAS SUWANWELA
452163	DILOK PHUVANANDH
645824	CHARAS SUWANWELA
756894	CHOTE PARNICHKULI
976582	THUMANNOON YONGCHAIYUDH

24	List of all passenger' IDs who were born before 2000-01-01.	<pre>SELECT Psg_BD AS "Passenger's Date of Birth", Psg_ID AS 'Passenger ID' FROM passenger WHERE Psg_BD < "2000-01-01" ORDER BY Psg_BD ASC;</pre>
----	---	--

Passenger's Date of Birth	Passenger ID
1955-11-22	912345678
1995-04-05	345678912
1997-06-07	567891234
1998-02-03	123456789
1999-09-09	789123456
1999-10-20	891234567

25	List of all the tickets that have an expiration date after 2019.	<pre>SELECT YEAR(Expiration_Date) AS "Expiration Year", Ticket_ID AS "Ticket ID" FROM ticket WHERE YEAR(Expiration_Date) > "2019" ORDER BY YEAR(Expiration_Date);</pre>
----	--	--

Expiration Year	Ticket ID
2020	100987654321
2020	112223333344
2021	113430009898
2022	119876543210
2023	118765432190
2025	117654321980
2027	110000123456

26	List of passengers who are older than the average of all passenger ages (age > 26).	<pre>SELECT Psg_ID AS 'Passenger ID', Psg_BD AS "Passenger's Date of Birth", (YEAR(CURDATE()) - YEAR(Psg_BD)) AS</pre>
----	---	--

		<pre> Age FROM passenger WHERE (YEAR(CURDATE()) - YEAR(Psg_BD)) > (SELECT AVG((YEAR(CURDATE()) - YEAR(Psg_BD))) FROM passenger) ORDER BY (YEAR(CURDATE()) - YEAR(Psg_BD)) ASC ; </pre>																																	
		<table border="1"> <thead> <tr> <th>Passenger ID</th> <th>Passenger's Date of Birth</th> <th>Age</th> </tr> </thead> <tbody> <tr> <td>345678912</td> <td>1995-04-05</td> <td>27</td> </tr> <tr> <td>912345678</td> <td>1955-11-22</td> <td>67</td> </tr> </tbody> </table>	Passenger ID	Passenger's Date of Birth	Age	345678912	1995-04-05	27	912345678	1955-11-22	67																								
Passenger ID	Passenger's Date of Birth	Age																																	
345678912	1995-04-05	27																																	
912345678	1955-11-22	67																																	
27	List of all the passenger's IDs that purchased the single journey ticket.	<pre> SELECT o.Order_No AS 'Order No', m.Psg_ID AS 'Passenger ID', t.Ticket_Type AS 'Ticket type' FROM setorder o INNER JOIN make m ON m.Order_No = o.Order_No INNER JOIN ticket t ON t.Order_No = o.Order_No WHERE Ticket_Type = "Single Journey" ORDER BY o.Order_No ASC ; </pre>																																	
		<table border="1"> <thead> <tr> <th>Order No</th> <th>Passenger ID</th> <th>Ticket type</th> </tr> </thead> <tbody> <tr><td>01234567</td><td>891234567</td><td>Single Journey</td></tr> <tr><td>21035412</td><td>123456789</td><td>Single Journey</td></tr> <tr><td>32041574</td><td>345678912</td><td>Single Journey</td></tr> <tr><td>34567812</td><td>234567891</td><td>Single Journey</td></tr> <tr><td>34681205</td><td>891234567</td><td>Single Journey</td></tr> <tr><td>45678123</td><td>345678912</td><td>Single Journey</td></tr> <tr><td>66795214</td><td>912345678</td><td>Single Journey</td></tr> <tr><td>67812345</td><td>567891234</td><td>Single Journey</td></tr> <tr><td>87654321</td><td>789123456</td><td>Single Journey</td></tr> <tr><td>98745031</td><td>456789123</td><td>Single Journey</td></tr> </tbody> </table>	Order No	Passenger ID	Ticket type	01234567	891234567	Single Journey	21035412	123456789	Single Journey	32041574	345678912	Single Journey	34567812	234567891	Single Journey	34681205	891234567	Single Journey	45678123	345678912	Single Journey	66795214	912345678	Single Journey	67812345	567891234	Single Journey	87654321	789123456	Single Journey	98745031	456789123	Single Journey
Order No	Passenger ID	Ticket type																																	
01234567	891234567	Single Journey																																	
21035412	123456789	Single Journey																																	
32041574	345678912	Single Journey																																	
34567812	234567891	Single Journey																																	
34681205	891234567	Single Journey																																	
45678123	345678912	Single Journey																																	
66795214	912345678	Single Journey																																	
67812345	567891234	Single Journey																																	
87654321	789123456	Single Journey																																	
98745031	456789123	Single Journey																																	
28	List the orders that have a higher total price than the average total price of all orders.	<pre> SELECT m.Order_No AS "Order ID", TotalPrice AS "Total Price" FROM make m LEFT OUTER JOIN invoice i ON m.Order_No = i.Order_No WHERE TotalPrice > (SELECT AVG(TotalPrice) FROM invoice) ORDER BY TotalPrice DESC; </pre>																																	

Order ID	Total Price
91420512	1435.00
87064259	1246.00
21035412	1244.00
76543210	1059.00
32041574	1000.00
65412743	1000.00
01234567	920.00
71025406	837.00
45678123	784.00
01476421	748.00
12345678	665.00
34567812	637.00
87654321	600.00

29	The range of ticket quantity that was purchased each year.	<pre>SELECT YEAR(Inv_DateTime) AS Year, CONCAT("[", "MIN(Ticket_Quantity),", ",MAX(Ticket_Quantity),"]) AS `Quantity range` FROM invoice i INNER JOIN invoiceline il ON i.INV_No = il.INV_No GROUP BY YEAR(Inv_DateTime) ORDER BY YEAR(Inv_DateTime) ASC ;</pre>
----	--	---

Year	Quantity range
2010	[1 , 4]
2011	[1 , 3]
2012	[1 , 3]
2013	[3 , 10]
2014	[4 , 14]
2015	[1 , 12]
2016	[1 , 5]
2017	[2 , 2]
2018	[2 , 5]
2019	[1 , 6]
2020	[1 , 10]
2021	[2 , 9]
2022	[1 , 8]

30	List the top 5 invoices that contain the highest ticket quantities.	<pre>SELECT DISTINCT il.INV_No AS "Invoice No" ,SUM(Ticket_Quantity) AS "Ticket Quantity" FROM invoiceline il LEFT OUTER JOIN invoice i ON il.INV_No = i.INV_No GROUP BY il.INV_No ORDER BY SUM(Ticket_Quantity) DESC</pre>
----	---	---

		LIMIT 5;
Invoice No	Ticket Quantity	
123-4-567890-1234-94	24	
236-9-510564-7789-56	23	
177-9-567890-2348-56	22	
142-2-632105-1123-62	20	
456-4-567890-1234-56	16	

31	List the most popular occupation among the passengers.	<pre>SELECT Psg_Occupation AS "Occupation", COUNT(Psg_Occupation) AS "Total of passengers" FROM passenger GROUP BY Psg_Occupation ORDER BY COUNT(Psg_Occupation) DESC LIMIT 1;</pre>
----	--	--

Occupation	Total of passengers
Student	2

32	List of the ticket sellers (Staff ID, Counter ID, First Name, and Last Name) that are close.	<pre>SELECT Staff_ID AS "Staff ID", Counter_ID AS "Counter ID", CONCAT(FirstName, " ", LastName) AS "Full Name" FROM ticketseller WHERE Tk_Status = "C";</pre>
----	--	--

Staff ID	Counter ID	Full Name
221034	N07	CHARAS SUWANWELA
258246	S10	PRADIST SAKSRI
354876	S12	ATT KETUSINGHA
578421	W01	PREECHA RUKSPOLMUANG

33	List of the total number of orders that were made between 6.00 AM to 10.00 AM.	<pre>SELECT CONCAT(HOUR(Order_DateTime),":00") AS "Time", COUNT(Order_No) AS "Total of order" FROM setorder WHERE HOUR(Order_DateTime) BETWEEN "06:00:00" AND "10:00:00" GROUP BY HOUR(Order_DateTime) ORDER BY HOUR(Order_DateTime) ASC;</pre>
----	--	---

Time	Total of order
6:00	1
7:00	1
8:00	1
9:00	1
10:00	1

34	The average total price of all orders.	SELECT ROUND(AVG(TotalPrice),2) AS "Average total price" FROM invoice;
----	--	---

Average total price

589.17

35	List of the top 5 invoices that have the lowest value of the total price.	SELECT Inv_No AS "Invoice ID", TotalPrice AS "Total Price", Inv_DateTime AS "Invoice Date-Time", Order_No AS "Order Number" FROM invoice ORDER BY TotalPrice ASC LIMIT 5;
----	---	--

Invoice ID	Total Price	Invoice Date-Time	Order Number
898-5-984562-2348-42	16.00	2011-11-23 06:57:59	23456781
978-9-445472-1234-56	175.00	2018-02-12 16:18:33	34167513
312-5-745212-6325-75	189.00	2012-10-26 09:49:40	56781234
155-9-567890-1234-56	223.00	2018-07-06 21:25:00	14702150
188-9-324517-1234-56	231.00	2020-04-09 20:16:11	31542012

36	List of the invoices that showed the purchase history of only Single Journey Ticket but not Rabbit Card.	SELECT DISTINCT il.INV_No AS "Invoice No", Inv_DateTime AS "Invoice Date-Time" FROM invoiceline il RIGHT OUTER JOIN invoice i ON il.INV_No = i.INV_No WHERE il.INV_No NOT IN (SELECT il.INV_No FROM invoiceline il LEFT OUTER JOIN invoice i ON il.INV_No = i.INV_No WHERE Ticket_Type = "Rabbit card");
----	--	--

Invoice No	Invoice Date-Time
123-4-567890-1234-94	2014-08-31 15:23:00
142-2-632105-1123-62	2013-10-30 13:37:44
155-9-567890-1234-56	2018-07-06 21:25:00
188-9-324517-1234-56	2020-04-09 20:16:11
312-5-745212-6325-75	2012-10-26 09:49:40
875-6-887542-6357-05	2021-03-12 17:14:44
898-5-984562-2348-42	2011-11-23 06:57:59
978-9-445472-1234-56	2018-02-12 16:18:33

37	List the invoices that showed the purchase history of only Rabbit Card but not Single Journey Ticket.	<pre>SELECT DISTINCT il.INV_No AS "Invoice No", Inv_DateTime AS "Invoice Date-Time" FROM invoiceline il RIGHT OUTER JOIN invoice i ON il.INV_No = i.INV_No WHERE il.INV_No NOT IN (SELECT il.INV_No FROM invoiceline il LEFT OUTER JOIN invoice i ON il.INV_No = i.INV_No WHERE Ticket_Type = "Single Journey");</pre>
----	---	--

Invoice No	Invoice Date-Time
122-7-324517-1234-71	2016-08-03 18:22:11
144-7-567890-2348-56	2017-06-05 20:22:11
553-3-845302-3320-34	2012-10-28 11:37:00
669-9-313326-4452-56	2019-12-14 13:12:00
854-6-984210-6849-89	2013-09-29 12:36:33
966-9-336521-1234-56	2018-01-13 15:11:22

38	List of the top 3 ticket sellers who sold the most tickets.	<pre>SELECT CONCAT(FirstName, " ", LastName) AS `Staff Name`, t.Staff_ID AS 'Staff ID', COUNT(Ticket_ID) AS 'Number of ticket' FROM ticket t INNER JOIN ticketseller ts ON t.Staff_ID = ts.Staff_ID GROUP BY t.Staff_ID ORDER BY COUNT(Ticket_ID) DESC LIMIT 3 ;</pre>
----	---	--

Staff Name	Staff ID	Number of ticket
------------	----------	------------------

JESADA NIMMANNITYA	134589	6
CHOTE PARNICKULI	756894	3
DILOK PHUVANANDH	452163	3

39	List of the average quantity of ticket that was purchased each year.	<pre> SELECT YEAR(Order_DateTime) AS "Year", ROUND(AVG(Ticket_Quantity)) AS "Average Ticket Quantity" FROM setorder s LEFT OUTER JOIN invoice i ON s.Order_No = i.Order_No LEFT OUTER JOIN invoice il ON i.INV_No = il.INV_No GROUP BY YEAR(Order_DateTime) ORDER BY YEAR(Order_DateTime) ASC; </pre>
----	--	---

Year	Average Ticket Quantity
------	-------------------------

2010	2
2011	2
2012	2
2013	8
2014	9
2015	4
2016	3
2017	2
2018	3
2019	4
2020	4
2021	5
2022	4

40	List of the top 5 order numbers that bought the highest quantity of the Rabbit card.	<pre> SELECT m.Order_No AS "Order ID", Ticket_Quantity AS "Ticket Quantity", Subtotal_Price AS "Ticket Price" FROM make m INNER JOIN invoice i ON m.Order_No = i.Order_No LEFT OUTER JOIN invoice il ON i.INV_No = il.INV_No WHERE Ticket_Type = "Rabbit card" GROUP BY i.Order_No ORDER BY Ticket_Quantity DESC LIMIT 5 ; </pre>
----	--	---

Order ID	Ticket Quantity	Ticket Price
65412743	5	1000.00
32041574	5	1000.00
76543210	4	800.00
71025406	4	800.00
87654321	3	600.00

References

1. “Company's profile,” รถไฟฟ้าบีทีเอส (BTS Skytrain). [Online]. Available: <https://www.bts.co.th/eng/info/info-history.html>.
2. “Our history,” *BTS Group Holdings*. [Online]. Available: <https://www.btsgroup.co.th/en/about/our-history>.
3. Norbert Braun, “Railway station”, digital image, *Unsplash*, [Online] Available: https://unsplash.com/photos/5RZqk8_omlA.
4. ThaimaaOpas, “BTS SKYTRAIN in Bangkok”, *digital image, Unsplash*, [Online] Available: https://unsplash.com/photos/IHNDE5Cb_RI.
5. ThaimaaOpas, “BTS skytrain public transport in Bangkok”, digital image, *Unsplash*, [Online] Available: <https://unsplash.com/photos/yXYT9dnEEPY>.
6. “Rabbit Group,” *Rabbit*. [Online]. Available: <https://rabbit.co.th/condition-of-issue>.
7. “BTS SkyTrain Route Map”, *digital image, btsapp1*, [Online]. Available: https://btsapp1.bts.co.th/WebApplication//Warehouse/Map/070264233236140121173519SystemMap_25Jan21_active.jpg.