**Department of Computer & Information Sciences**

|  |  |
| --- | --- |
| **ASSESSMENT SUBMISSION** | |
| **Module Title:** | Advanced Databases |
| **Module Code:** | KL7011 |
| **Academic Year / Semester:** | 2025-256 / Semester 1 |
| **Module Tutor / Email (all queries):** | Akhtar Ali [akhtar.ali@northumbria.ac.uk](mailto:akhtar.ali@northumbria.ac.uk) |
| **% Weighting (to overall module):** | 60% |
| **Assessment Title:** | Assignment 1: individual work |
| **Date of Handout to Students:** | 20th October 2025 |
| **Mechanism for Handout:** | Module Blackboard Site |
| **Deadline for Submission Attempt by Students:** | Thu 27/11/2025 @ 16:00 GMT |
| **Mechanism for Submission:** | Document upload to Module Blackboard Site |
| **Submission Format / Word Count** | Please upload your written report as a single PDF document |
| **Date by which Work, Feedback and Marks will be returned:** | 31st December 2025 |
| **Mechanism for return of Feedback and Marks:** | Mark and individual written feedback will be uploaded to the Module Site on Blackboard. For further queries please email module tutor. |
| **Student ID** | W24041293 |
| **Oracle Username** | **advDBusr98** |
| **Student Name** | Sumit Malviya |

**Assignment # 1**

**Personalising your SQL output/prompt**

Before executing any **SQL code** for this assignment, you should personalise your SQL output / prompt by running SET SQLPROMPT “UniversityUserName > ”, i.e., *double-quote* followed by your UniversityUserName followed by > and then a ***space*** and *double-quote* as shown in the screenshot below:

*A screenshot of a computer

AI-generated content may be incorrect.*

If you are using Oracle Live or Oracle FreeSQL then you should prefix your table names with your UniversityUserName, e.g., Create Table w123456\_emp;

**Assignment Questions/Tasks**

**Part 1 (40 marks)**

This part is based on the NORTHERNTOURS scenario as described in the Appendix.

1. *Using entity-relationship (ER) OR enhanced entity-relationship (EER) modelling, produce a conceptual design of the database for supporting the information system of the NORTHERNTOURS company.*

(20 marks)

Answer Part 1 A: Insert below your ER or EER Diagram

**A diagram of a company

AI-generated content may be incorrect.**

Provide below any assumptions you made whilst choosing entity and relationship types. For example, in the Company database scenario, one may assume that an employee can only manage one department; similarly, in a flights-booking-system, one may assume that an aeroplane assigned to a flight leg instance is not in repair/maintenance status and that the pilot allocated to fly the aeroplane has a valid licence. Up to 2 Marks will be subtracted if assumptions are not stated or otherwise not adequate.

In designing this EER model I made the following assumptions:

I used an enhanced ER (EER) instead of a simple ER because I wanted to show extra detail. In my EER, **PERSON** is a supertype and **DRIVER** and **CLERK** are non-overlapping subtypes, and the primary key person\_id is also used as the primary key and foreign key in driver and clerk. This avoids repeating basic details like name, phone and email.

For each **SCHEDULE** , I assume there is exactly one **COACH** and one **DRIVER**, and both are available and valid at the departure time. A **ticket** always refers to one schedule and represents one seat, so there is no overbooking. A **reservation** is always made by one **CUSTOMER** through one **AGENT**, but the same customer can make many reservations over time and can use different agents on different bookings.

I also assume that **ROUTE\_STOP** and **STOP\_MEAL** are real entities created from many-to-many relationships, with their own attributes such as stop\_sequence, duration\_min, price and vegetarian. Changes to a schedule are stored in **SCHEDULE\_CHANGE**, where each change belongs to exactly one schedule, and a schedule can have many changes. Finally, I added a separate **TRANSACTION\_** entity so that payment details are stored separately from ticket details, with at most one **TRANSACTION**\_ per ticket.

Provide below a brief discussion on why your conceptual design (ERD/EERD) is fit for the given scenario, whether or not it covers the full scope and supports the required functionality of the system. Up to 3 Marks will be subtracted if discussion is either missing or not adequate.

My EER diagram is designed to match the main needs of the NorthernTours coach booking scenario. The PERSON–DRIVER–CLERK structure lets me keep common PERSON details in one place, while still showing that DRIVERS and CLERKS have different roles and attributes. AGENT and CUSTOMER are linked to reservation, ticket and TRANSACTION\_, so the company can track who made each booking, which seats were sold, and how each ticket was paid.

ROUTE, STOP and ROUTE\_STOP describe each coach route with an ordered list of stops. MEAL and STOP\_MEAL record which meals are available at which stops and at what price. SCHEDULE then joins a specific COACH and DRIVER to a route on a given date and time, which supports planning future trips. SCHEDULE\_CHANGE allows the company to record important changes to a schedule over time, such as delays or cancellations, together with the reason and time of the change.

Overall, the design supports the key tasks in the scenario -> defining routes, assigning coaches and drivers, creating reservations, issuing tickets, taking payments and tracking schedule changes. The conceptual design is fit for the required system.

1. *Convert the ER / EER diagram from Part 1(A) to produce a logical relational schema using ER / EER to relational mapping.*

(12 marks)

Answer Part 1 B: Provide below your Logical Relational Design/Schema (8 marks)

**A diagram of a diagram

AI-generated content may be incorrect.**

* **Strong entities → tables** Each main entity in the EER becomes its own table: ROUTE, STOP, MEAL, COACH, SCHEDULE, SCHEDULE\_CHANGE, RESERVATION, AGENT, CUSTOMER, PERSON, TICKET, TRANSACTION\_
* **1:M relationships → foreign key on the “many” side** Examples:
  1. SCHEDULE.ROUTE\_ID → ROUTE.ROUTE\_ID
  2. SCHEDULE.COACH\_ID → COACH.COACH\_ID
  3. SCHEDULE.DRIVER\_ID → DRIVER.PERSON\_ID
  4. TICKET.SCHEDULE\_ID → SCHEDULE.SCHEDULE\_ID
  5. TICKET.RESERVATION\_ID → RESERVATION.RESERVATION\_ID
  6. RESERVATION.AGENT\_ID → AGENT.AGENT\_ID
  7. RESERVATION.CUSTOMER\_ID → CUSTOMER.CUSTOMER\_ID
  8. SCHEDULE\_CHANGE.SCHEDULE\_ID → SCHEDULE.SCHEDULE\_ID
  9. TRANSACTION\_.TICKET\_ID → TICKET.TICKET\_ID
* **M:N relationships → junction tables** M:N relationships are mapped to separate tables with composite keys:
  1. ROUTE\_STOP(ROUTE\_STOP\_ID, ROUTE\_ID, STOP\_SEQUENCE, STOP\_ID, DURATION\_MIN)
  2. STOP\_MEAL(STOP\_MEAL\_ID, STOP\_ID, MEAL\_ID, PRICE, VEGETARIAN)
* **Ordered component relationship (Route–Stops)** ROUTE\_STOP stores the ordered list of stops for each route:
  1. Primary key: (ROUTE\_ID, STOP\_SEQUENCE) keeps the sequence of stops.
  2. UNIQUE(ROUTE\_ID, STOP\_ID) avoids listing the same stop twice on the same route.
* **Location-specific products (meals at stops)** STOP\_MEAL(STOP\_ID, MEAL\_ID, PRICE, VEGETARIAN) links meals to individual stops:
  1. Primary key: (STOP\_ID, MEAL\_ID) ensures each stop–meal combination appears at most once.
* **Subtype specialisation (disjoint, PK = FK)** DRIVER(PERSON\_ID, …) and CLERK(PERSON\_ID, …) are subtypes of PERSON:
  1. PERSON\_ID is both the primary key and a foreign key to PERSON(PERSON\_ID).
  2. This follows the disjoint specialisation shown in the EER.
* **1:1 business rule (one payment per ticket)** In TRANSACTION\_, the column TICKET\_ID is a foreign key to TICKET **and** is declared UNIQUE. This enforces at most one transaction (one payment) for each ticket.

• **Overbooking prevention (design note)** SCHEDULE.SEATS\_AVAILABLE records how many seats remain on a schedule. Each TICKET represents one seat, so overbooking can be prevented at application level or with triggers based on this value.

Answer Part 1 B: Provide below your Data Dictionary (in a tabular form and must be presented as text rather than an image or picture) (4 marks)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table** | **Column** | **Data Type** | **PK** | **FK (References)** | **Nullable** | **Default / Check / Unique Notes** | **Description** |
| PERSON | person\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique identifier for each person |
| PERSON | first\_name | VARCHAR2(50) |  | – | N |  | Person’s first name |
| PERSON | last\_name | VARCHAR2(50) |  | – | N |  | Person’s last name |
| PERSON | phone | VARCHAR2(30) |  | – | Y |  | Contact phone number (optional) |
| PERSON | email | VARCHAR2(120) |  | – | Y |  | Contact email (optional) |
| DRIVER | person\_id | NUMBER(10) | Y | PERSON(person\_id) | N | PK = FK | Person acting as a driver |
| DRIVER | license\_number | VARCHAR2(30) |  | – | N | UNIQUE | Driving licence number |
| DRIVER | hire\_date | DATE |  | – | Y |  | Date driver was hired (optional) |
| DRIVER | status | VARCHAR2(20) |  | – | Y |  | Employment / availability status |
| CLERK | person\_id | NUMBER(10) | Y | PERSON(person\_id) | N | PK = FK | Person acting as a clerk |
| CLERK | office | VARCHAR2(50) |  | – | Y |  | Office location (optional) |
| AGENT | agent\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique identifier for each agent |
| AGENT | name | VARCHAR2(100) |  | – | N |  | Agent name |
| AGENT | email | VARCHAR2(120) |  | – | Y |  | Agent email (optional) |
| AGENT | telephone | VARCHAR2(30) |  | – | Y |  | Agent phone (optional) |
| AGENT | status | VARCHAR2(20) |  | – | Y |  | Agent status (optional) |
| AGENT | address | VARCHAR2(200) |  | – | Y |  | Agent address (optional) |
| CUSTOMER | customer\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique identifier for each customer |
| CUSTOMER | name | VARCHAR2(100) |  | – | N |  | Customer name |
| CUSTOMER | email | VARCHAR2(120) |  | – | Y |  | Customer email (optional) |
| CUSTOMER | telephone | VARCHAR2(30) |  | – | Y |  | Customer phone (optional) |
| CUSTOMER | address | VARCHAR2(200) |  | – | Y |  | Customer address (optional) |
| COACH | coach\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique identifier for each coach |
| COACH | registration\_no | VARCHAR2(20) |  | – | N | UNIQUE | Vehicle registration number |
| COACH | capacity | NUMBER(3) |  | – | N | CHECK (capacity > 0) | Number of passenger seats |
| COACH | make | VARCHAR2(50) |  | – | Y |  | Manufacturer / make (optional) |
| COACH | manufacture\_yr | NUMBER(4) |  | – | Y |  | Year of manufacture (optional) |
| COACH | last\_service\_date | DATE |  | – | Y |  | Date of last service (optional) |
| COACH | wifi\_available | CHAR(1) |  | – | Y | CHECK (wifi\_available IN ('Y','N')) | Whether Wi-Fi is available (optional) |
| COACH | coach\_model | VARCHAR2(50) |  | – | Y |  | Model description (optional) |
| ROUTE | route\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique identifier for each route |
| ROUTE | route\_name | VARCHAR2(100) |  | – | N | UNIQUE | Name / label for the route |
| ROUTE | description | VARCHAR2(200) |  | – | Y |  | Route description (optional) |
| ROUTE | total\_distance | NUMBER(6,1) |  | – | Y |  | Total distance in km (optional) |
| ROUTE | duration | NUMBER(5) |  | – | Y |  | Planned duration in minutes (optional) |
| STOP | stop\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique identifier for each stop |
| STOP | stop\_name | VARCHAR2(100) |  | – | N |  | Name of the stop |
| STOP | city | VARCHAR2(100) |  | – | Y |  | City / town (optional) |
| STOP | address | VARCHAR2(200) |  | – | Y |  | Address or description (optional) |
| STOP | parking\_available | CHAR(1) |  | – | Y | CHECK (parking\_available IN ('Y','N')) | Whether parking is available (optional) |
| STOP | facilities | VARCHAR2(200) |  | – | Y |  | Facilities description (optional) |
| MEAL | meal\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique identifier for each meal |
| MEAL | meal\_name | VARCHAR2(100) |  | – | N |  | Meal name |
| MEAL | meal\_type | VARCHAR2(50) |  | – | Y |  | Meal type/category (optional) |
| MEAL | description | VARCHAR2(200) |  | – | Y |  | Meal description (optional) |
| ROUTE\_STOP | route\_id | NUMBER(10) | Y | ROUTE(route\_id) | N | Part of composite PK | Route for this stop |
| ROUTE\_STOP | stop\_sequence | NUMBER(3) | Y | – | N | Part of composite PK; defines order along the route | Sequence number of stop in route |
| ROUTE\_STOP | stop\_id | NUMBER(10) |  | STOP(stop\_id) | N | UNIQUE (route\_id, stop\_id) within a route | Linked stop on the route |
| ROUTE\_STOP | duration\_min | NUMBER(5) |  | – | Y |  | Planned minutes from previous stop (optional) |
| STOP\_MEAL | stop\_id | NUMBER(10) | Y | STOP(stop\_id) | N | Part of composite PK | Stop where meal is available |
| STOP\_MEAL | meal\_id | NUMBER(10) | Y | MEAL(meal\_id) | N | Part of composite PK | Meal offered at that stop |
| STOP\_MEAL | price | NUMBER(6,2) |  | – | Y |  | Local price at that stop (optional) |
| STOP\_MEAL | vegetarian | CHAR(1) |  | – | Y | CHECK (vegetarian IN ('Y','N')) | Whether meal is vegetarian (optional) |
| SCHEDULE | schedule\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique identifier for each schedule |
| SCHEDULE | route\_id | NUMBER(10) |  | ROUTE(route\_id) | N |  | Route used by the schedule |
| SCHEDULE | coach\_id | NUMBER(10) |  | COACH(coach\_id) | N |  | Coach assigned |
| SCHEDULE | driver\_id | NUMBER(10) |  | DRIVER(person\_id) | N |  | Driver assigned |
| SCHEDULE | departure\_datetime | DATE |  | – | N |  | Planned departure date and time |
| SCHEDULE | ticket\_price | NUMBER(6,2) |  | – | N |  | Standard ticket price |
| SCHEDULE | seats\_available | NUMBER(3) |  | – | Y | CHECK (seats\_available >= 0) | Remaining seats (optional) |
| SCHEDULE | service\_status | VARCHAR2(20) |  | – | Y | CHECK (service\_status IN ('scheduled','delayed','cancelled','completed')) | Service status (optional) |
| SCHEDULE\_CHANGE | change\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique schedule change record |
| SCHEDULE\_CHANGE | schedule\_id | NUMBER(10) |  | SCHEDULE(schedule\_id) | N |  | Schedule that was changed |
| SCHEDULE\_CHANGE | change\_time | DATE |  | – | N |  | When the change was made |
| SCHEDULE\_CHANGE | change\_type | VARCHAR2(20) |  | – | N |  | Type of change (e.g. delay, cancel) |
| SCHEDULE\_CHANGE | reason | VARCHAR2(200) |  | – | Y |  | Reason for change (optional) |
| RESERVATION | reservation\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique reservation identifier |
| RESERVATION | agent\_id | NUMBER(10) |  | AGENT(agent\_id) | N |  | Agent who created the reservation |
| RESERVATION | customer\_id | NUMBER(10) |  | CUSTOMER(customer\_id) | N |  | Customer for the reservation |
| RESERVATION | reservation\_date | DATE |  | – | N |  | Date reservation was made |
| RESERVATION | requested\_seats | NUMBER(3) |  | – | N | CHECK (requested\_seats > 0) | Number of seats requested |
| RESERVATION | reservation\_status | VARCHAR2(20) |  | – | Y |  | Reservation status (optional) |
| TICKET | ticket\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique ticket identifier |
| TICKET | reservation\_id | NUMBER(10) |  | RESERVATION(reservation\_id) | N |  | Reservation this ticket belongs to |
| TICKET | schedule\_id | NUMBER(10) |  | SCHEDULE(schedule\_id) | N |  | Schedule for the journey |
| TICKET | issue\_date | DATE |  | – | N |  | Date ticket was issued |
| TICKET | seat\_allocation | VARCHAR2(10) |  | – | N |  | Seat number |
| TICKET | price | NUMBER(6,2) |  | – | N |  | Final ticket price |
| TICKET | status | VARCHAR2(20) |  | – | Y |  | Ticket status (optional) |
| TRANSACTION\_ | transaction\_id | NUMBER(10) | Y | – | N | GENERATED AS IDENTITY | Unique payment transaction identifier |
| TRANSACTION\_ | ticket\_id | NUMBER(10) |  | TICKET(ticket\_id) | N | UNIQUE | Ticket being paid for (one transaction per ticket) |
| TRANSACTION\_ | transaction\_date | DATE |  | – | N |  | Date of payment |
| TRANSACTION\_ | payment\_method | VARCHAR2(20) |  | – | N |  | Payment method (card, cash, etc.) |
| TRANSACTION\_ | paid\_amount | NUMBER(8,2) |  | – | N | CHECK (paid\_amount >= 0) | Amount paid |
| TRANSACTION\_ | reference\_number | VARCHAR2(30) |  | – | Y |  | External reference (optional) |
| TRANSACTION\_ | transaction\_status | VARCHAR2(20) |  | – | Y |  | Status of the transaction (optional) |

Provide below a short statement about what naming convention you used for different elements of your logical relational schema/data dictionary and justifiy your choice. Up to 2 Marks will be subtracted if naming convention is either missing not justified or if it is not adequate.

**Naming convention used and justification:**

In my logical relational database model and data dictionary, all table names I refer to must be in UPPER\_SNAKE\_CASE (i.e. ROUTE, SCHEDULE, STOP\_MEAL, TRANSACTION\_). This enables the tables to be easily recognizable within SQL code and gives uniformity throughout the entire design.

I write columns in lower\_snake\_case (which would be things like route\_id, schedule\_id, ticket\_price). Table’s primary keys are in the table \_id (example: route\_id, customer\_id), and foreign key have names as their respective referenced primary key. This itself tells which column joins to which table and this also prevents one getting confused of where writing joins. In general, it is simple enough to read and understanding, and is good for an oracle database design.

Provide below a short statement about confirming whether or not your relations in the logical schema are in 3rd Normal Form. Up to 2 Marks will be subtracted if such a statement is missing / not all relations in 3rd Normal Form or if it is not adequate.

**Normalisation and 3rd Normal Form:**

I normalised the logical relational schema such that all relations are in 3NF. For each table there is a well-defined key (it is obvious the primary key), and all non-key attributes depend on the full key, not less, and nothing else. Repeating groups are not present, nor are multi-valued attributes: instead, many-to-many relationships (e.g., between route and stop, and stop and meal) are handled by the junction tables ROUTE\_STOP table and STOP\_MEAL.

In the case of a composite key (as in ROUTE\_STOP and STOP\_MEAL), that mutual non-key dependency holds for all columns given by the complete composite key.  
  
Transitive Dependencies can be removed by decomposing the relation that it depends on with different concepts, e.g — Keeping PERSON in a separate table (not driver and clerks) DRIVER, CLERK, CUSTOMER, AGENT, SCHEDULE, TICKET, TRANSACTION\_. On the basis of these checks, I think my logical schema is in 3rd N.F.

1. *Based on your logical design from Part 1 (B) and the information available in the scenario, produce an SQL script file using Oracle 11g/12c/higher.*

(8 marks)

Answer Part 1 C: Provide SQL DDL Script file contents as Text not image or picture (i.e., the SQL code for creating / altering your Tables / Constraints etc)

-- NORTHERNTOURS – CREATE TABLES

CREATE TABLE PERSON (

person\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

first\_name VARCHAR2(50) NOT NULL,

last\_name VARCHAR2(50) NOT NULL,

phone VARCHAR2(30),

email VARCHAR2(120)

);

CREATE TABLE DRIVER (

person\_id NUMBER(10),

license\_number VARCHAR2(30) NOT NULL,

hire\_date DATE,

status VARCHAR2(20)

);

CREATE TABLE CLERK (

person\_id NUMBER(10),

office VARCHAR2(50)

);

CREATE TABLE AGENT (

agent\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

name VARCHAR2(100) NOT NULL,

email VARCHAR2(120),

telephone VARCHAR2(30),

status VARCHAR2(20),

address VARCHAR2(200)

);

CREATE TABLE CUSTOMER (

customer\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

name VARCHAR2(100) NOT NULL,

email VARCHAR2(120),

telephone VARCHAR2(30),

address VARCHAR2(200)

);

CREATE TABLE COACH (

coach\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

registration\_no VARCHAR2(20) NOT NULL,

capacity NUMBER(3) NOT NULL,

make VARCHAR2(50),

manufacture\_yr NUMBER(4),

last\_service\_date DATE,

wifi\_available CHAR(1),

coach\_model VARCHAR2(50)

);

CREATE TABLE ROUTE (

route\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

route\_name VARCHAR2(100) NOT NULL,

description VARCHAR2(200),

total\_distance NUMBER(6,1),

duration NUMBER(5)

);

CREATE TABLE STOP (

stop\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

stop\_name VARCHAR2(100) NOT NULL,

city VARCHAR2(100),

address VARCHAR2(200),

parking\_available CHAR(1),

facilities VARCHAR2(200)

);

CREATE TABLE MEAL (

meal\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

meal\_name VARCHAR2(100) NOT NULL,

meal\_type VARCHAR2(50),

description VARCHAR2(200)

);

CREATE TABLE ROUTE\_STOP (

route\_id NUMBER(10) NOT NULL,

stop\_sequence NUMBER(3) NOT NULL,

stop\_id NUMBER(10) NOT NULL,

duration\_min NUMBER(5)

);

CREATE TABLE STOP\_MEAL (

stop\_id NUMBER(10) NOT NULL,

meal\_id NUMBER(10) NOT NULL,

price NUMBER(6,2),

vegetarian CHAR(1)

);

CREATE TABLE SCHEDULE (

schedule\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

route\_id NUMBER(10) NOT NULL,

coach\_id NUMBER(10) NOT NULL,

driver\_id NUMBER(10) NOT NULL,

departure\_datetime DATE NOT NULL,

ticket\_price NUMBER(6,2) NOT NULL,

seats\_available NUMBER(3),

service\_status VARCHAR2(20)

);

CREATE TABLE SCHEDULE\_CHANGE (

change\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

schedule\_id NUMBER(10) NOT NULL,

change\_time DATE NOT NULL,

change\_type VARCHAR2(20) NOT NULL,

reason VARCHAR2(200)

);

CREATE TABLE RESERVATION (

reservation\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

agent\_id NUMBER(10) NOT NULL,

customer\_id NUMBER(10) NOT NULL,

reservation\_date DATE NOT NULL,

requested\_seats NUMBER(3) NOT NULL,

reservation\_status VARCHAR2(20)

);

CREATE TABLE TICKET (

ticket\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

reservation\_id NUMBER(10) NOT NULL,

schedule\_id NUMBER(10) NOT NULL,

issue\_date DATE NOT NULL,

seat\_allocation VARCHAR2(10) NOT NULL,

price NUMBER(6,2) NOT NULL,

status VARCHAR2(20)

);

CREATE TABLE TRANSACTION\_ (

transaction\_id NUMBER(10) GENERATED ALWAYS AS IDENTITY,

ticket\_id NUMBER(10) NOT NULL,

transaction\_date DATE NOT NULL,

payment\_method VARCHAR2(20) NOT NULL,

paid\_amount NUMBER(8,2) NOT NULL,

reference\_number VARCHAR2(30),

transaction\_status VARCHAR2(20)

);

-- NORTHERNTOURS – CONSTRAINTS (ALTER TABLE):

ALTER TABLE PERSON

ADD CONSTRAINT pk\_person PRIMARY KEY (person\_id);

ALTER TABLE DRIVER

ADD CONSTRAINT pk\_driver PRIMARY KEY (person\_id);

ALTER TABLE CLERK

ADD CONSTRAINT pk\_clerk PRIMARY KEY (person\_id);

ALTER TABLE AGENT

ADD CONSTRAINT pk\_agent PRIMARY KEY (agent\_id);

ALTER TABLE CUSTOMER

ADD CONSTRAINT pk\_customer PRIMARY KEY (customer\_id);

ALTER TABLE COACH

ADD CONSTRAINT pk\_coach PRIMARY KEY (coach\_id);

ALTER TABLE ROUTE

ADD CONSTRAINT pk\_route PRIMARY KEY (route\_id);

ALTER TABLE STOP

ADD CONSTRAINT pk\_stop PRIMARY KEY (stop\_id);

ALTER TABLE MEAL

ADD CONSTRAINT pk\_meal PRIMARY KEY (meal\_id);

ALTER TABLE ROUTE\_STOP

ADD CONSTRAINT pk\_route\_stop PRIMARY KEY (route\_id, stop\_sequence);

ALTER TABLE STOP\_MEAL

ADD CONSTRAINT pk\_stop\_meal PRIMARY KEY (stop\_id, meal\_id);

ALTER TABLE SCHEDULE

ADD CONSTRAINT pk\_schedule PRIMARY KEY (schedule\_id);

ALTER TABLE SCHEDULE\_CHANGE

ADD CONSTRAINT pk\_schedule\_change PRIMARY KEY (change\_id);

ALTER TABLE RESERVATION

ADD CONSTRAINT pk\_reservation PRIMARY KEY (reservation\_id);

ALTER TABLE TICKET

ADD CONSTRAINT pk\_ticket PRIMARY KEY (ticket\_id);

ALTER TABLE TRANSACTION\_

ADD CONSTRAINT pk\_transaction PRIMARY KEY (transaction\_id);

ALTER TABLE DRIVER

ADD CONSTRAINT fk\_driver\_person

FOREIGN KEY (person\_id)

REFERENCES PERSON(person\_id);

ALTER TABLE CLERK

ADD CONSTRAINT fk\_clerk\_person

FOREIGN KEY (person\_id)

REFERENCES PERSON(person\_id);

ALTER TABLE ROUTE\_STOP

ADD CONSTRAINT fk\_route\_stop\_route

FOREIGN KEY (route\_id)

REFERENCES ROUTE(route\_id);

ALTER TABLE ROUTE\_STOP

ADD CONSTRAINT fk\_route\_stop\_stop

FOREIGN KEY (stop\_id)

REFERENCES STOP(stop\_id);

ALTER TABLE STOP\_MEAL

ADD CONSTRAINT fk\_stop\_meal\_stop

FOREIGN KEY (stop\_id)

REFERENCES STOP(stop\_id);

ALTER TABLE STOP\_MEAL

ADD CONSTRAINT fk\_stop\_meal\_meal

FOREIGN KEY (meal\_id)

REFERENCES MEAL(meal\_id);

ALTER TABLE SCHEDULE

ADD CONSTRAINT fk\_schedule\_route

FOREIGN KEY (route\_id)

REFERENCES ROUTE(route\_id);

ALTER TABLE SCHEDULE

ADD CONSTRAINT fk\_schedule\_coach

FOREIGN KEY (coach\_id)

REFERENCES COACH(coach\_id);

ALTER TABLE SCHEDULE

ADD CONSTRAINT fk\_schedule\_driver

FOREIGN KEY (driver\_id)

REFERENCES DRIVER(person\_id);

ALTER TABLE SCHEDULE\_CHANGE

ADD CONSTRAINT fk\_schedule\_change\_schedule

FOREIGN KEY (schedule\_id)

REFERENCES SCHEDULE(schedule\_id);

ALTER TABLE RESERVATION

ADD CONSTRAINT fk\_reservation\_agent

FOREIGN KEY (agent\_id)

REFERENCES AGENT(agent\_id);

ALTER TABLE RESERVATION

ADD CONSTRAINT fk\_reservation\_customer

FOREIGN KEY (customer\_id)

REFERENCES CUSTOMER(customer\_id);

ALTER TABLE TICKET

ADD CONSTRAINT fk\_ticket\_reservation

FOREIGN KEY (reservation\_id)

REFERENCES RESERVATION(reservation\_id);

ALTER TABLE TICKET

ADD CONSTRAINT fk\_ticket\_schedule

FOREIGN KEY (schedule\_id)

REFERENCES SCHEDULE(schedule\_id);

ALTER TABLE TRANSACTION\_

ADD CONSTRAINT fk\_transaction\_ticket

FOREIGN KEY (ticket\_id)

REFERENCES TICKET(ticket\_id);

ALTER TABLE DRIVER

ADD CONSTRAINT uq\_driver\_license UNIQUE (license\_number);

ALTER TABLE COACH

ADD CONSTRAINT uq\_coach\_reg UNIQUE (registration\_no);

ALTER TABLE ROUTE

ADD CONSTRAINT uq\_route\_name UNIQUE (route\_name);

ALTER TABLE ROUTE\_STOP

ADD CONSTRAINT uq\_route\_stop UNIQUE (route\_id, stop\_id);

ALTER TABLE TRANSACTION\_

ADD CONSTRAINT uq\_transaction\_ticket UNIQUE (ticket\_id);

ALTER TABLE COACH

ADD CONSTRAINT ck\_coach\_capacity

CHECK (capacity > 0);

ALTER TABLE COACH

ADD CONSTRAINT ck\_coach\_wifi

CHECK (wifi\_available IN ('Y','N') OR wifi\_available IS NULL);

ALTER TABLE STOP

ADD CONSTRAINT ck\_stop\_parking

CHECK (parking\_available IN ('Y','N') OR parking\_available IS NULL);

ALTER TABLE STOP\_MEAL

ADD CONSTRAINT ck\_stop\_meal\_veg

CHECK (vegetarian IN ('Y','N') OR vegetarian IS NULL);

ALTER TABLE SCHEDULE

ADD CONSTRAINT ck\_schedule\_seats

CHECK (seats\_available >= 0 OR seats\_available IS NULL);

ALTER TABLE SCHEDULE

ADD CONSTRAINT ck\_schedule\_status

CHECK (

service\_status IN ('scheduled','delayed','cancelled','completed')

OR service\_status IS NULL

);

ALTER TABLE RESERVATION

ADD CONSTRAINT ck\_reservation\_seats

CHECK (requested\_seats > 0);

ALTER TABLE TRANSACTION\_

ADD CONSTRAINT ck\_transaction\_amount

CHECK (paid\_amount >= 0);

Answer Part 1 C: SQL DDL Output (e.g., SPOOL file contents or output you got when you executed your above SQL Table Creation code, this should show the SQL code as well as its output). Make sure the output is a screenshot / image or picture and NOT a simple TEXT as it may increase your Turn-it-in similarity score. If output is missing, 2 marks will be deducted from the above 8 marks. If output is incomplete or inadequate or misleading, then adequate marks up to a max of 2 will be deducted.

A screenshot of a computer screen

AI-generated content may be incorrect.

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**SQL DDL Constraints (ALTER TABLE statements):**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Part 2 (15 marks)**

This part is based on your answer / solution to Part 1 (design and implementation of the database) for the NORTHERNTOURS scenario.

*(A) Populate a subset of the relations of t@ "E:\NorthernTours\Scripts\ddl.sql"he database with sample data for answering the queries in Par 2 (B), i.e., you should generate your own dummy data and load it into the NORTHERNTOURS database, consider 5 to 10 rows for each table in your subset and enough data to see meaningful output for the queries below*.

(7 marks)

Answer Part 2 A: Provide SQL code below as Text not image or picture for populating the subset of relations of the above database.

**-- NORTHERNTOURS – SAMPLE DATA** :  
  
INSERT INTO person (first\_name, last\_name, phone, email) VALUES

('Aditi', 'Sharma', '+44 7700 900111', 'aditi.sharma@northerntours.co.uk');

INSERT INTO person (first\_name, last\_name, phone, email) VALUES

('Emily', 'Watson', '+44 7700 900222', 'emily.watson@northerntours.co.uk');

INSERT INTO person (first\_name, last\_name, phone, email) VALUES

('Michael', 'Carter', '+1 212 555 0101', 'michael.carter@northerntours.com');

INSERT INTO person (first\_name, last\_name, phone, email) VALUES

('Rohan', 'Patel', '+91 98 7654 3210','rohan.patel@northerntours.co.uk');

INSERT INTO person (first\_name, last\_name, phone, email) VALUES

('Sarah', 'Johnson', '+1 415 555 0202', 'sarah.johnson@northerntours.com');

INSERT INTO person (first\_name, last\_name, phone, email) VALUES

('Liam', 'Brown', '+44 7700 900333', 'liam.brown@northerntours.co.uk');

INSERT INTO person (first\_name, last\_name, phone, email) VALUES

('Olivia', 'Davis', '+44 7700 900444', 'olivia.davis@northerntours.co.uk');

INSERT INTO person (first\_name, last\_name, phone, email) VALUES

('Noah', 'Wilson', '+44 7700 900555', 'noah.wilson@northerntours.co.uk');

INSERT INTO person (first\_name, last\_name, phone, email) VALUES

('Zara', 'Khan', '+44 7700 900666', 'zara.khan@northerntours.co.uk');

INSERT INTO driver (person\_id, license\_number, hire\_date, status)

SELECT person\_id, 'UK-DRV-1001', TO\_DATE('2020-03-15','YYYY-MM-DD'), 'active'

FROM person

WHERE first\_name = 'Aditi' AND last\_name = 'Sharma';

INSERT INTO driver (person\_id, license\_number, hire\_date, status)

SELECT person\_id, 'UK-DRV-1002', TO\_DATE('2019-06-01','YYYY-MM-DD'), 'active'

FROM person

WHERE first\_name = 'Emily' AND last\_name = 'Watson';

INSERT INTO driver (person\_id, license\_number, hire\_date, status)

SELECT person\_id, 'US-DRV-2001', TO\_DATE('2018-09-10','YYYY-MM-DD'), 'active'

FROM person

WHERE first\_name = 'Michael' AND last\_name = 'Carter';

INSERT INTO driver (person\_id, license\_number, hire\_date, status)

SELECT person\_id, 'UK-DRV-3001', TO\_DATE('2021-01-05','YYYY-MM-DD'), 'active'

FROM person

WHERE first\_name = 'Liam' AND last\_name = 'Brown';

INSERT INTO driver (person\_id, license\_number, hire\_date, status)

SELECT person\_id, 'UK-DRV-4001', TO\_DATE('2022-02-10','YYYY-MM-DD'), 'active'

FROM person

WHERE first\_name = 'Noah' AND last\_name = 'Wilson';

INSERT INTO driver (person\_id, license\_number, hire\_date, status)

SELECT person\_id, 'UK-DRV-5001', TO\_DATE('2023-04-18','YYYY-MM-DD'), 'active'

FROM person

WHERE first\_name = 'Zara' AND last\_name = 'Khan';

INSERT INTO agent (name, email, telephone, status, address) VALUES

('Northern Travels Newcastle', 'info@northerntravels.co.uk',

'+44 191 555 0100', 'active', 'Newcastle upon Tyne, UK');

INSERT INTO agent (name, email, telephone, status, address) VALUES

('Durham City Breaks', 'sales@durhambreaks.co.uk',

'+44 191 555 0200', 'active', 'Durham, UK');

INSERT INTO agent (name, email, telephone, status, address) VALUES

('Global Trips London', 'contact@globaltrips.co.uk',

'+44 20 7000 1234', 'active', 'London, UK');

INSERT INTO agent (name, email, telephone, status, address) VALUES

('Sunshine Tours Mumbai', 'hello@sunshinetours.in',

'+91 22 4000 5678', 'active', 'Mumbai, India');

INSERT INTO agent (name, email, telephone, status, address) VALUES

('Pacific Getaways New York', 'bookings@pacificgetaways.com',

'+1 212 555 0909', 'active', 'New York, USA');

INSERT INTO agent (name, email, telephone, status, address) VALUES

('Highland Explorer Tours', 'info@highlandexplorer.co.uk',

'+44 131 555 0300', 'active', 'Edinburgh, UK');

INSERT INTO customer (name, email, telephone, address) VALUES

('Priya Singh', 'priya.singh@example.com', '+44 7700 111111',

'Gosforth, Newcastle upon Tyne, UK');

INSERT INTO customer (name, email, telephone, address) VALUES

('Oliver Green', 'oliver.green@example.com', '+44 7700 222222',

'Jesmond, Newcastle upon Tyne, UK');

INSERT INTO customer (name, email, telephone, address) VALUES

('Sophia Miller', 'sophia.miller@example.com', '+1 415 555 0303',

'San Francisco, USA');

INSERT INTO customer (name, email, telephone, address) VALUES

('Arjun Rao', 'arjun.rao@example.com', '+91 98 1111 2222',

'Bangalore, India');

INSERT INTO customer (name, email, telephone, address) VALUES

('Chloe Anderson', 'chloe.anderson@example.com', '+44 7700 333333',

'Durham, UK');

INSERT INTO customer (name, email, telephone, address) VALUES

('Ethan Walker', 'ethan.walker@example.com', '+44 7700 444444',

'York, UK');

INSERT INTO customer (name, email, telephone, address) VALUES

('Mia Patel', 'mia.patel@example.com', '+44 7700 555555',

'Sunderland, UK');

INSERT INTO coach (registration\_no, capacity, make, manufacture\_yr,

last\_service\_date, wifi\_available, coach\_model)

VALUES ('NT01 NCL', 50, 'Volvo', 2020,

TO\_DATE('2024-06-01','YYYY-MM-DD'), 'Y', 'Volvo 9700');

INSERT INTO coach (registration\_no, capacity, make, manufacture\_yr,

last\_service\_date, wifi\_available, coach\_model)

VALUES ('NT02 DUR', 40, 'Mercedes', 2019,

TO\_DATE('2024-05-15','YYYY-MM-DD'), 'Y', 'Tourismo');

INSERT INTO coach (registration\_no, capacity, make, manufacture\_yr,

last\_service\_date, wifi\_available, coach\_model)

VALUES ('NT03 EDI', 55, 'Scania', 2018,

TO\_DATE('2024-04-20','YYYY-MM-DD'), 'N', 'Irizar i6');

INSERT INTO coach (registration\_no, capacity, make, manufacture\_yr,

last\_service\_date, wifi\_available, coach\_model)

VALUES ('NT04 YORK', 45, 'MAN', 2021,

TO\_DATE('2024-07-10','YYYY-MM-DD'), 'Y', 'Lion Coach');

INSERT INTO coach (registration\_no, capacity, make, manufacture\_yr,

last\_service\_date, wifi\_available, coach\_model)

VALUES ('NT05 NCL2', 48, 'Volvo', 2022,

TO\_DATE('2024-07-20','YYYY-MM-DD'), 'Y', 'Volvo 9900');

INSERT INTO route (route\_name, description, total\_distance, duration) VALUES

('Newcastle - Berwick Day Tour',

'Scenic coastal route from Newcastle to Berwick-upon-Tweed',

90.0, 120);

INSERT INTO route (route\_name, description, total\_distance, duration) VALUES

('Newcastle - Durham Commuter',

'Quick commuter route between Newcastle and Durham',

30.0, 45);

INSERT INTO route (route\_name, description, total\_distance, duration) VALUES

('Edinburgh - Berwick Coastal',

'Coastal service from Edinburgh to Berwick-upon-Tweed',

60.0, 90);

INSERT INTO route (route\_name, description, total\_distance, duration) VALUES

('York - Newcastle Express',

'Express coach from York to Newcastle',

130.0, 150);

INSERT INTO route (route\_name, description, total\_distance, duration) VALUES

('Newcastle - Edinburgh Direct',

'Direct service from Newcastle to Edinburgh',

160.0, 180);

INSERT INTO route (route\_name, description, total\_distance, duration) VALUES

('Durham - York Shuttle',

'Shuttle service between Durham and York',

95.0, 110);

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities) VALUES

('Newcastle Coach Station', 'Newcastle upon Tyne',

'Neville Street, Newcastle upon Tyne', 'Y',

'Toilets; Cafe; Waiting area');

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities) VALUES

('Berwick-upon-Tweed Bus Station', 'Berwick-upon-Tweed',

'Walkergate, Berwick-upon-Tweed', 'N',

'Sheltered stands');

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities) VALUES

('Durham Bus Station', 'Durham',

'North Road, Durham', 'Y',

'Shops; Toilets');

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities) VALUES

('Morpeth Bus Station', 'Morpeth',

'Bridge Street, Morpeth', 'N',

'Cafe; Seating');

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities) VALUES

('Edinburgh Bus Station', 'Edinburgh',

'Elder Street, Edinburgh', 'Y',

'Shops; Toilets; WiFi');

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities) VALUES

('York Coach Station', 'York',

'Station Road, York', 'Y',

'Cafe; Toilets; Waiting room');

INSERT INTO route\_stop (route\_id, stop\_sequence, stop\_id, duration\_min)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Newcastle - Berwick Day Tour'),

1,

(SELECT stop\_id FROM stop

WHERE stop\_name = 'Newcastle Coach Station'),

0

);

INSERT INTO route\_stop (route\_id, stop\_sequence, stop\_id, duration\_min)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Newcastle - Berwick Day Tour'),

2,

(SELECT stop\_id FROM stop

WHERE stop\_name = 'Morpeth Bus Station'),

20

);

INSERT INTO route\_stop (route\_id, stop\_sequence, stop\_id, duration\_min)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Newcastle - Berwick Day Tour'),

3,

(SELECT stop\_id FROM stop

WHERE stop\_name = 'Berwick-upon-Tweed Bus Station'),

0

);

INSERT INTO route\_stop (route\_id, stop\_sequence, stop\_id, duration\_min)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Newcastle - Durham Commuter'),

1,

(SELECT stop\_id FROM stop

WHERE stop\_name = 'Newcastle Coach Station'),

0

);

INSERT INTO route\_stop (route\_id, stop\_sequence, stop\_id, duration\_min)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Newcastle - Durham Commuter'),

2,

(SELECT stop\_id FROM stop

WHERE stop\_name = 'Durham Bus Station'),

0

);

INSERT INTO route\_stop (route\_id, stop\_sequence, stop\_id, duration\_min)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Edinburgh - Berwick Coastal'),

1,

(SELECT stop\_id FROM stop

WHERE stop\_name = 'Edinburgh Bus Station'),

0

);

INSERT INTO route\_stop (route\_id, stop\_sequence, stop\_id, duration\_min)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Edinburgh - Berwick Coastal'),

2,

(SELECT stop\_id FROM stop

WHERE stop\_name = 'Berwick-upon-Tweed Bus Station'),

0

);

INSERT INTO route\_stop (route\_id, stop\_sequence, stop\_id, duration\_min)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'York - Newcastle Express'),

1,

(SELECT stop\_id FROM stop

WHERE stop\_name = 'York Coach Station'),

0

);

INSERT INTO route\_stop (route\_id, stop\_sequence, stop\_id, duration\_min)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'York - Newcastle Express'),

2,

(SELECT stop\_id FROM stop

WHERE stop\_name = 'Newcastle Coach Station'),

0

);

INSERT INTO schedule (route\_id, coach\_id, driver\_id,

departure\_datetime, ticket\_price,

seats\_available, service\_status)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Newcastle - Berwick Day Tour'),

(SELECT coach\_id FROM coach

WHERE registration\_no = 'NT01 NCL'),

(SELECT d.person\_id FROM driver d

JOIN person p ON d.person\_id = p.person\_id

WHERE p.first\_name = 'Aditi' AND p.last\_name = 'Sharma'),

TO\_DATE('2025-12-02 09:00','YYYY-MM-DD HH24:MI'),

25.00,

40,

'scheduled'

);

INSERT INTO schedule (route\_id, coach\_id, driver\_id,

departure\_datetime, ticket\_price,

seats\_available, service\_status)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Newcastle - Berwick Day Tour'),

(SELECT coach\_id FROM coach

WHERE registration\_no = 'NT01 NCL'),

(SELECT d.person\_id FROM driver d

JOIN person p ON d.person\_id = p.person\_id

WHERE p.first\_name = 'Emily' AND p.last\_name = 'Watson'),

TO\_DATE('2025-12-03 09:00','YYYY-MM-DD HH24:MI'),

25.00,

5,

'scheduled'

);

INSERT INTO schedule (route\_id, coach\_id, driver\_id,

departure\_datetime, ticket\_price,

seats\_available, service\_status)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Newcastle - Berwick Day Tour'),

(SELECT coach\_id FROM coach

WHERE registration\_no = 'NT02 DUR'),

(SELECT d.person\_id FROM driver d

JOIN person p ON d.person\_id = p.person\_id

WHERE p.first\_name = 'Michael' AND p.last\_name = 'Carter'),

TO\_DATE('2025-12-05 14:00','YYYY-MM-DD HH24:MI'),

27.00,

18,

'scheduled'

);

INSERT INTO schedule (route\_id, coach\_id, driver\_id,

departure\_datetime, ticket\_price,

seats\_available, service\_status)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Newcastle - Durham Commuter'),

(SELECT coach\_id FROM coach

WHERE registration\_no = 'NT02 DUR'),

(SELECT d.person\_id FROM driver d

JOIN person p ON d.person\_id = p.person\_id

WHERE p.first\_name = 'Emily' AND p.last\_name = 'Watson'),

TO\_DATE('2025-12-02 08:00','YYYY-MM-DD HH24:MI'),

15.00,

30,

'scheduled'

);

INSERT INTO schedule (route\_id, coach\_id, driver\_id,

departure\_datetime, ticket\_price,

seats\_available, service\_status)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'Edinburgh - Berwick Coastal'),

(SELECT coach\_id FROM coach

WHERE registration\_no = 'NT03 EDI'),

(SELECT d.person\_id FROM driver d

JOIN person p ON d.person\_id = p.person\_id

WHERE p.first\_name = 'Michael' AND p.last\_name = 'Carter'),

TO\_DATE('2025-12-04 10:00','YYYY-MM-DD HH24:MI'),

30.00,

10,

'scheduled'

);

INSERT INTO schedule (route\_id, coach\_id, driver\_id,

departure\_datetime, ticket\_price,

seats\_available, service\_status)

VALUES (

(SELECT route\_id FROM route

WHERE route\_name = 'York - Newcastle Express'),

(SELECT coach\_id FROM coach

WHERE registration\_no = 'NT04 YORK'),

(SELECT d.person\_id FROM driver d

JOIN person p ON d.person\_id = p.person\_id

WHERE p.first\_name = 'Liam' AND p.last\_name = 'Brown'),

TO\_DATE('2025-12-10 07:30','YYYY-MM-DD HH24:MI'),

20.00,

25,

'scheduled'

);

INSERT INTO reservation (agent\_id, customer\_id, reservation\_date,

requested\_seats, reservation\_status)

VALUES (

(SELECT agent\_id FROM agent

WHERE name = 'Northern Travels Newcastle'),

(SELECT customer\_id FROM customer

WHERE name = 'Priya Singh'),

TO\_DATE('2025-09-15','YYYY-MM-DD'),

2,

'confirmed'

);

INSERT INTO reservation (agent\_id, customer\_id, reservation\_date,

requested\_seats, reservation\_status)

VALUES (

(SELECT agent\_id FROM agent

WHERE name = 'Northern Travels Newcastle'),

(SELECT customer\_id FROM customer

WHERE name = 'Oliver Green'),

TO\_DATE('2025-09-16','YYYY-MM-DD'),

1,

'confirmed'

);

INSERT INTO reservation (agent\_id, customer\_id, reservation\_date,

requested\_seats, reservation\_status)

VALUES (

(SELECT agent\_id FROM agent

WHERE name = 'Northern Travels Newcastle'),

(SELECT customer\_id FROM customer

WHERE name = 'Sophia Miller'),

TO\_DATE('2025-10-01','YYYY-MM-DD'),

1,

'confirmed'

);

INSERT INTO reservation (agent\_id, customer\_id, reservation\_date,

requested\_seats, reservation\_status)

VALUES (

(SELECT agent\_id FROM agent

WHERE name = 'Durham City Breaks'),

(SELECT customer\_id FROM customer

WHERE name = 'Arjun Rao'),

TO\_DATE('2025-09-20','YYYY-MM-DD'),

2,

'confirmed'

);

INSERT INTO reservation (agent\_id, customer\_id, reservation\_date,

requested\_seats, reservation\_status)

VALUES (

(SELECT agent\_id FROM agent

WHERE name = 'Global Trips London'),

(SELECT customer\_id FROM customer

WHERE name = 'Chloe Anderson'),

TO\_DATE('2025-10-05','YYYY-MM-DD'),

1,

'confirmed'

);

INSERT INTO reservation (agent\_id, customer\_id, reservation\_date,

requested\_seats, reservation\_status)

VALUES (

(SELECT agent\_id FROM agent

WHERE name = 'Sunshine Tours Mumbai'),

(SELECT customer\_id FROM customer

WHERE name = 'Mia Patel'),

TO\_DATE('2025-09-01','YYYY-MM-DD'),

1,

'cancelled'

);

INSERT INTO ticket (reservation\_id, schedule\_id, issue\_date,

seat\_allocation, price, status)

VALUES (

(SELECT reservation\_id FROM reservation

WHERE reservation\_date = TO\_DATE('2025-09-15','YYYY-MM-DD')

AND customer\_id = (SELECT customer\_id FROM customer

WHERE name = 'Priya Singh')),

(SELECT schedule\_id FROM schedule s

JOIN route r ON s.route\_id = r.route\_id

WHERE r.route\_name = 'Newcastle - Berwick Day Tour'

AND s.departure\_datetime = TO\_DATE('2025-12-02 09:00',

'YYYY-MM-DD HH24:MI')),

TO\_DATE('2025-09-15','YYYY-MM-DD'),

'1A',

25.00,

'confirmed'

);

INSERT INTO ticket (reservation\_id, schedule\_id, issue\_date,

seat\_allocation, price, status)

VALUES (

(SELECT reservation\_id FROM reservation

WHERE reservation\_date = TO\_DATE('2025-09-15','YYYY-MM-DD')

AND customer\_id = (SELECT customer\_id FROM customer

WHERE name = 'Priya Singh')),

(SELECT schedule\_id FROM schedule s

JOIN route r ON s.route\_id = r.route\_id

WHERE r.route\_name = 'Newcastle - Berwick Day Tour'

AND s.departure\_datetime = TO\_DATE('2025-12-02 09:00',

'YYYY-MM-DD HH24:MI')),

TO\_DATE('2025-09-15','YYYY-MM-DD'),

'1B',

25.00,

'confirmed'

);

INSERT INTO ticket (reservation\_id, schedule\_id, issue\_date,

seat\_allocation, price, status)

VALUES (

(SELECT reservation\_id FROM reservation

WHERE reservation\_date = TO\_DATE('2025-09-16','YYYY-MM-DD')

AND customer\_id = (SELECT customer\_id FROM customer

WHERE name = 'Oliver Green')),

(SELECT schedule\_id FROM schedule s

JOIN route r ON s.route\_id = r.route\_id

WHERE r.route\_name = 'Newcastle - Berwick Day Tour'

AND s.departure\_datetime = TO\_DATE('2025-12-05 14:00',

'YYYY-MM-DD HH24:MI')),

TO\_DATE('2025-09-16','YYYY-MM-DD'),

'3C',

27.00,

'confirmed'

);

INSERT INTO ticket (reservation\_id, schedule\_id, issue\_date,

seat\_allocation, price, status)

VALUES (

(SELECT reservation\_id FROM reservation

WHERE reservation\_date = TO\_DATE('2025-10-01','YYYY-MM-DD')

AND customer\_id = (SELECT customer\_id FROM customer

WHERE name = 'Sophia Miller')),

(SELECT schedule\_id FROM schedule s

JOIN route r ON s.route\_id = r.route\_id

WHERE r.route\_name = 'York - Newcastle Express'

AND s.departure\_datetime = TO\_DATE('2025-12-10 07:30',

'YYYY-MM-DD HH24:MI')),

TO\_DATE('2025-10-01','YYYY-MM-DD'),

'5D',

20.00,

'confirmed'

);

INSERT INTO ticket (reservation\_id, schedule\_id, issue\_date,

seat\_allocation, price, status)

VALUES (

(SELECT reservation\_id FROM reservation

WHERE reservation\_date = TO\_DATE('2025-09-20','YYYY-MM-DD')

AND customer\_id = (SELECT customer\_id FROM customer

WHERE name = 'Arjun Rao')),

(SELECT schedule\_id FROM schedule s

JOIN route r ON s.route\_id = r.route\_id

WHERE r.route\_name = 'Newcastle - Durham Commuter'

AND s.departure\_datetime = TO\_DATE('2025-12-02 08:00',

'YYYY-MM-DD HH24:MI')),

TO\_DATE('2025-09-20','YYYY-MM-DD'),

'2A',

15.00,

'confirmed'

);

INSERT INTO ticket (reservation\_id, schedule\_id, issue\_date,

seat\_allocation, price, status)

VALUES (

(SELECT reservation\_id FROM reservation

WHERE reservation\_date = TO\_DATE('2025-09-20','YYYY-MM-DD')

AND customer\_id = (SELECT customer\_id FROM customer

WHERE name = 'Arjun Rao')),

(SELECT schedule\_id FROM schedule s

JOIN route r ON s.route\_id = r.route\_id

WHERE r.route\_name = 'Newcastle - Durham Commuter'

AND s.departure\_datetime = TO\_DATE('2025-12-02 08:00',

'YYYY-MM-DD HH24:MI')),

TO\_DATE('2025-09-20','YYYY-MM-DD'),

'2B',

15.00,

'confirmed'

);

INSERT INTO ticket (reservation\_id, schedule\_id, issue\_date,

seat\_allocation, price, status)

VALUES (

(SELECT reservation\_id FROM reservation

WHERE reservation\_date = TO\_DATE('2025-10-05','YYYY-MM-DD')

AND customer\_id = (SELECT customer\_id FROM customer

WHERE name = 'Chloe Anderson')),

(SELECT schedule\_id FROM schedule s

JOIN route r ON s.route\_id = r.route\_id

WHERE r.route\_name = 'Edinburgh - Berwick Coastal'

AND s.departure\_datetime = TO\_DATE('2025-12-04 10:00',

'YYYY-MM-DD HH24:MI')),

TO\_DATE('2025-10-05','YYYY-MM-DD'),

'4A',

30.00,

'confirmed'

);

INSERT INTO ticket (reservation\_id, schedule\_id, issue\_date,

seat\_allocation, price, status)

VALUES (

(SELECT reservation\_id FROM reservation

WHERE reservation\_date = TO\_DATE('2025-09-01','YYYY-MM-DD')

AND customer\_id = (SELECT customer\_id FROM customer

WHERE name = 'Mia Patel')),

(SELECT schedule\_id FROM schedule s

JOIN route r ON s.route\_id = r.route\_id

WHERE r.route\_name = 'Newcastle - Berwick Day Tour'

AND s.departure\_datetime = TO\_DATE('2025-12-03 09:00',

'YYYY-MM-DD HH24:MI')),

TO\_DATE('2025-09-01','YYYY-MM-DD'),

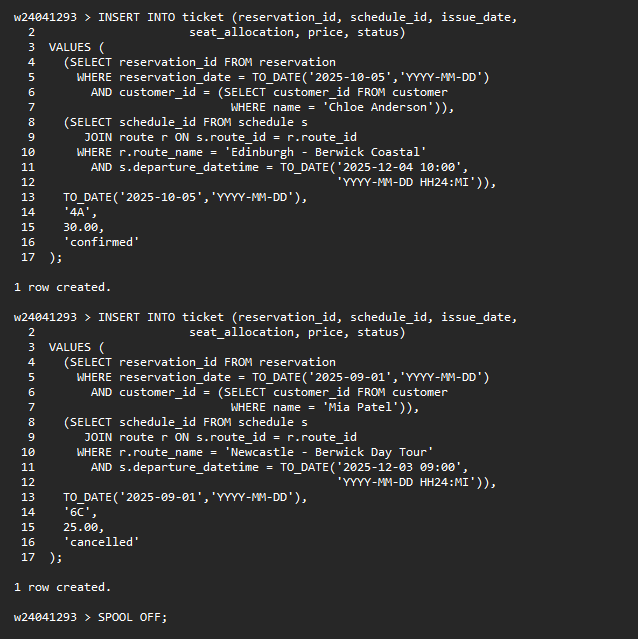
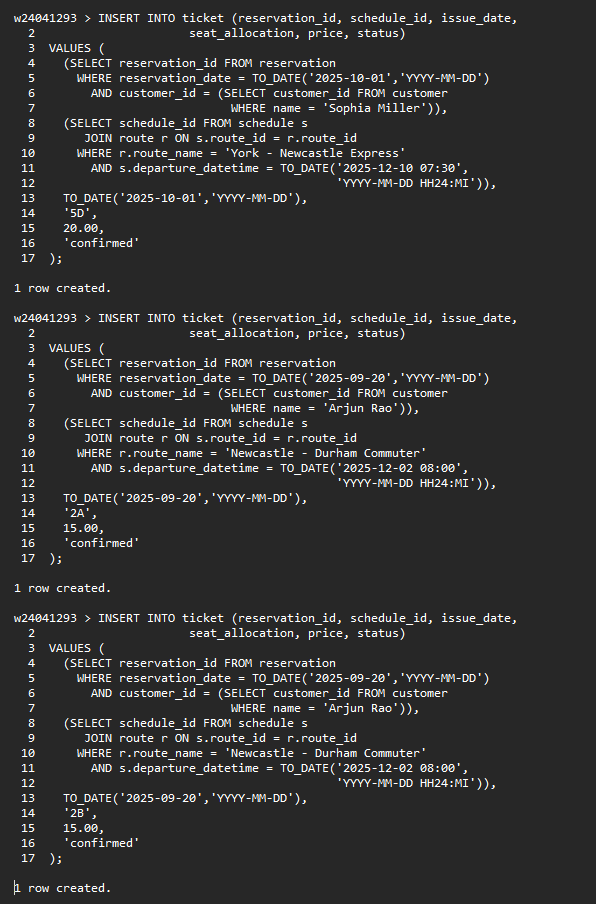
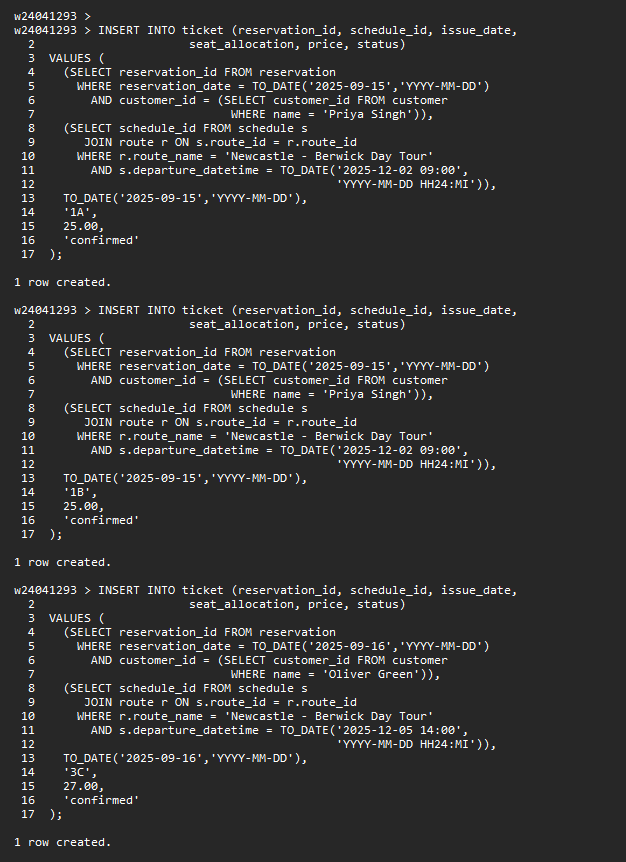
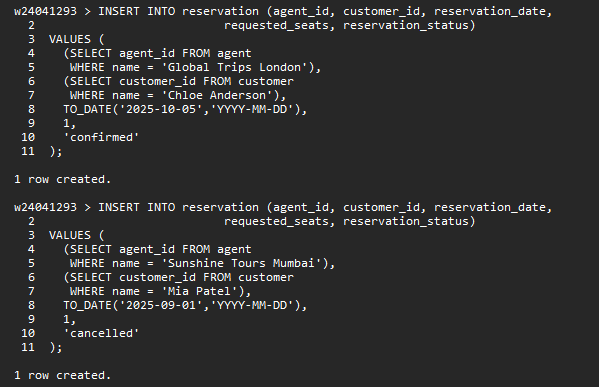
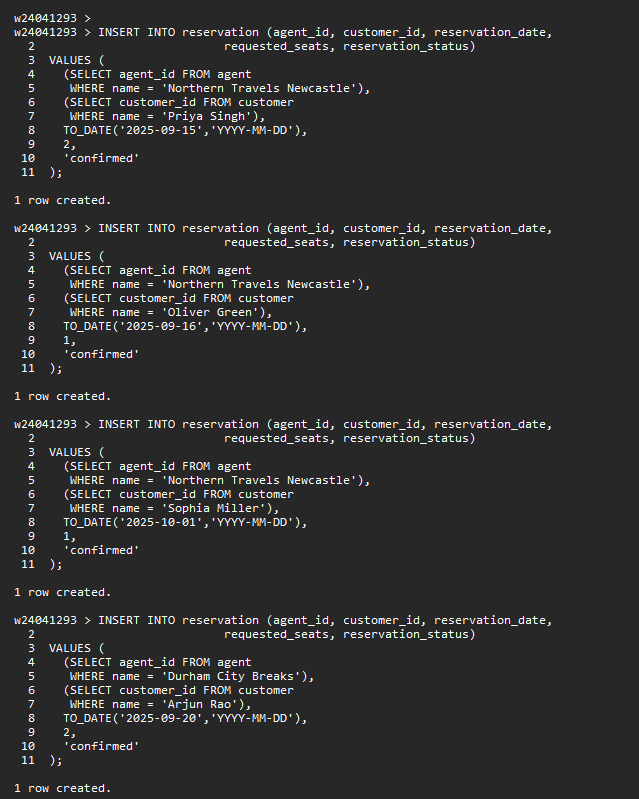
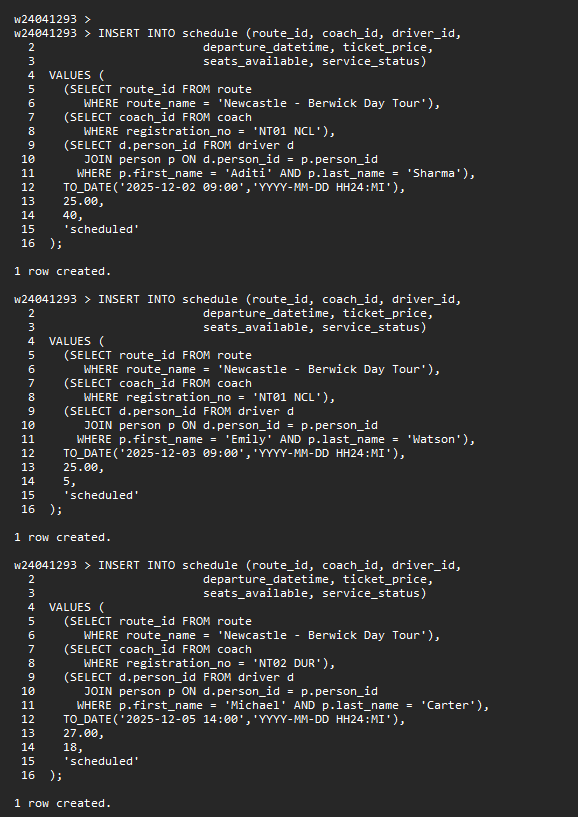
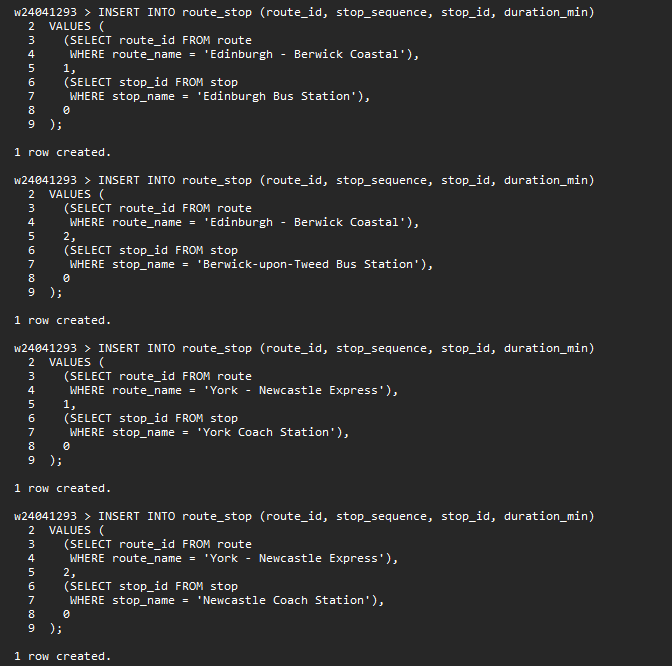
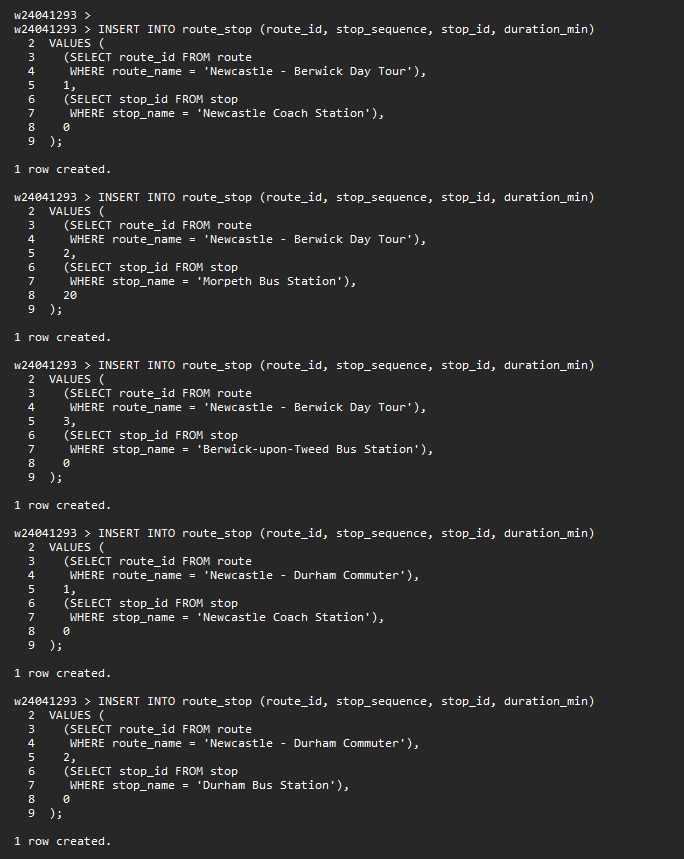
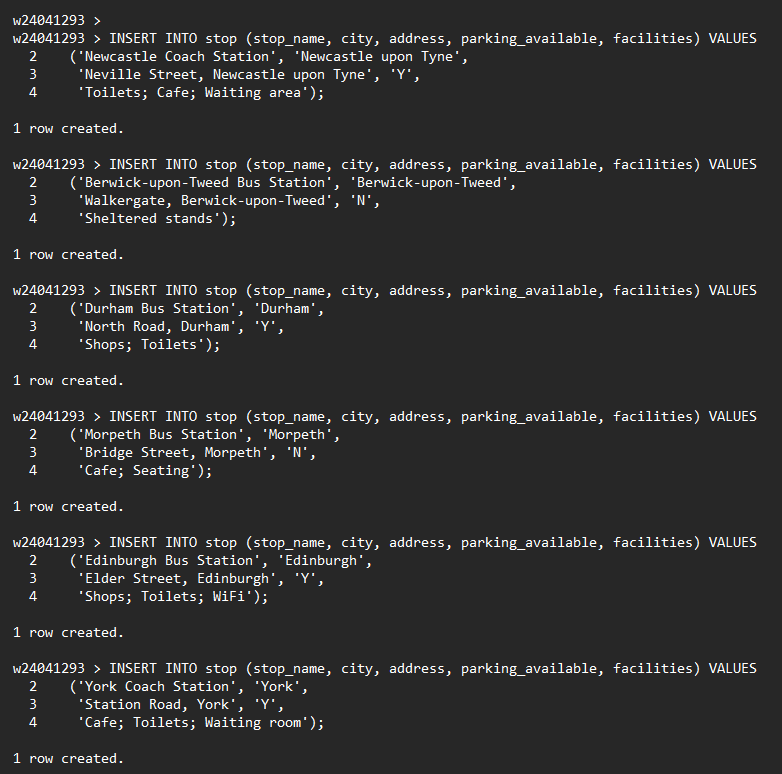
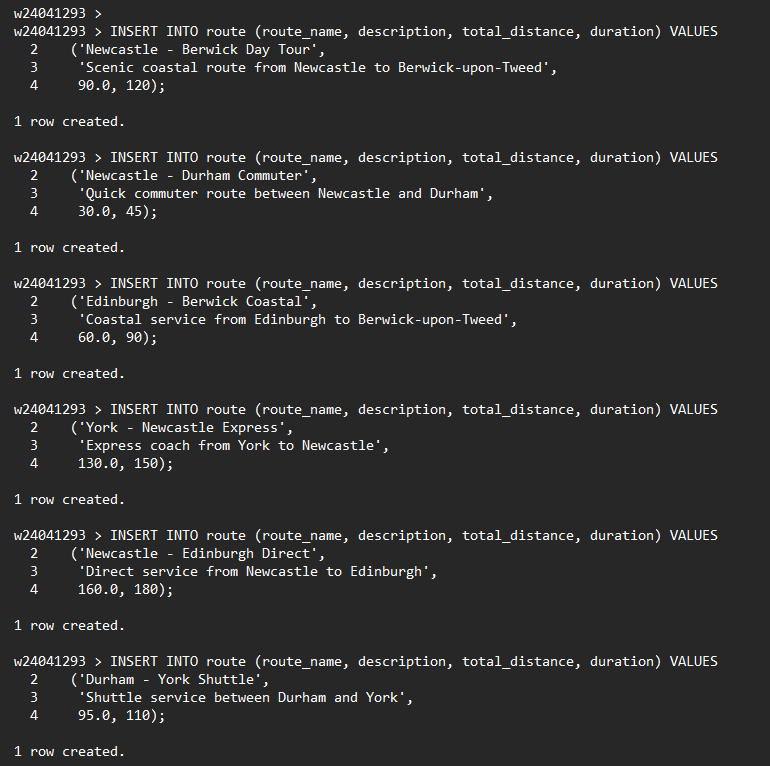
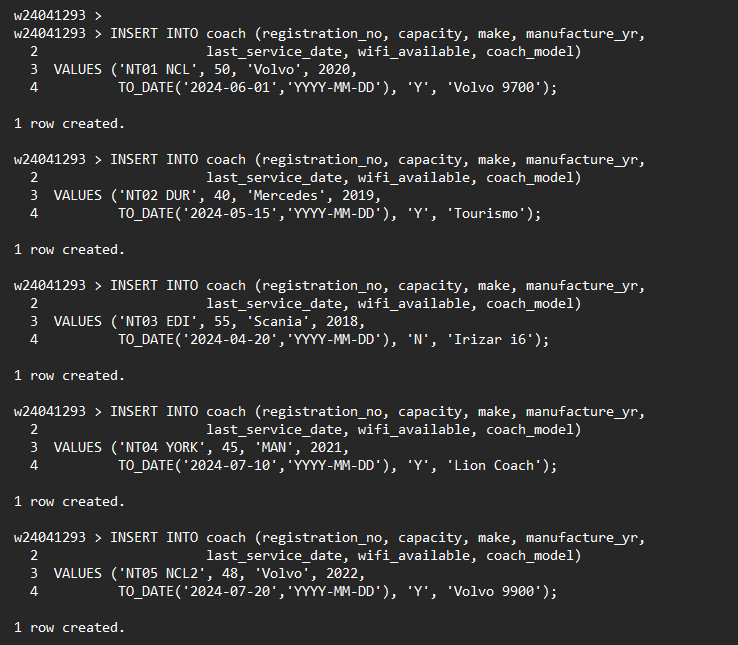
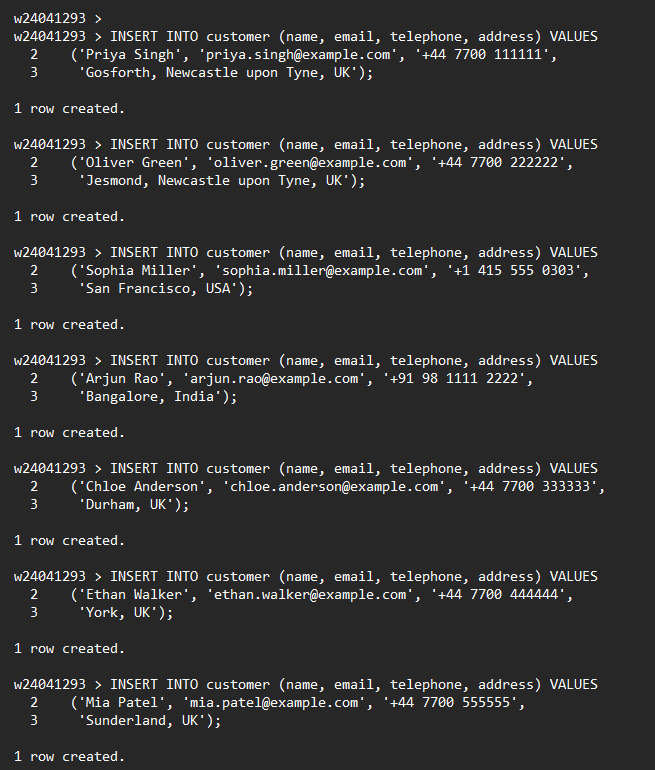
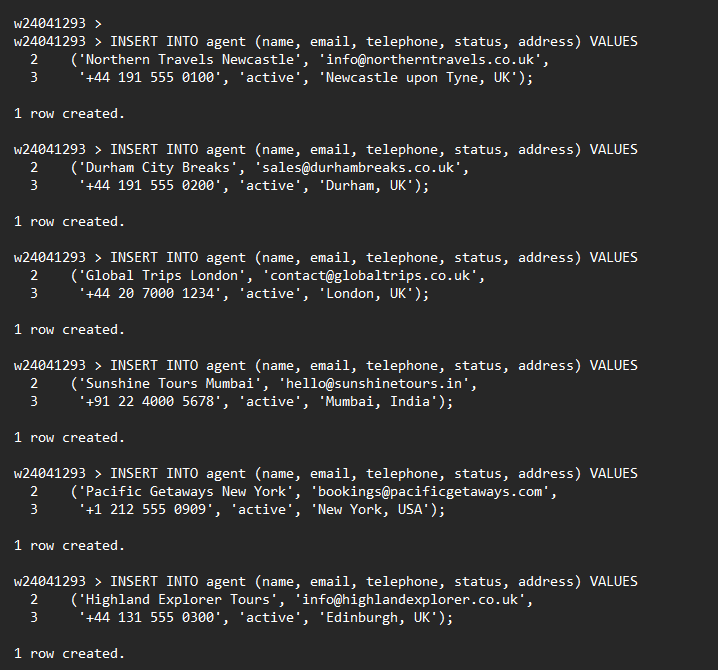
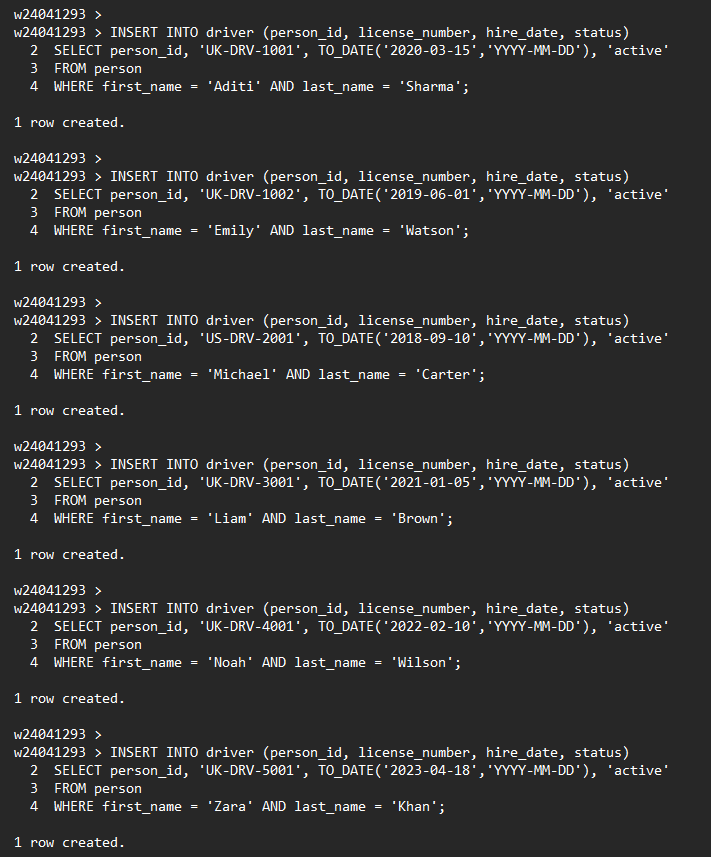
'6C',

25.00,

'cancelled'

);

Answer Part 2 A: Provide below output from running the above SQL code for populating the subset of your relational database (e.g., contents from Spool file or copy & paste of outputs from the SQL plus window). Make sure the output is a screenshot / image or picture and NOT a simple TEXT as it may increase your Turn-it-in similarity score

****

*(B) Answer the following queries (retrievals) using Relational Algebra and SQL.*

(8 marks)

1. Display details of schedules for travelling between Newcastle and Berwick-upon-Tweed with seven or more available seats in the next 14 days.

Provide Relational Algebra expression below:

**q1) Relational Algebra:**

I only use the tables needed: SCHEDULE and ROUTE.

1. **Pick the Newcastle–Berwick route:**

R1 = σ\_{route\_name = 'Newcastle - Berwick Day Tour'} (ROUTE)

2. **Pick schedules in the next 14 days with at least 7 seats:**

S1 = σ\_{seats\_available >= 7 AND

departure\_datetime >= TODAY AND

departure\_datetime <= TODAY + 14} (SCHEDULE)

3. **Join them and show the main details:**

Result = π\_{schedule\_id, route\_name, departure\_datetime,

ticket\_price, seats\_available}

( S1 ⋈\_{S1.route\_id = R1.route\_id} R1 )

Provide SQL query code as TEXT and output below (as image / screenshot / picture):

1. Display details of the travel agent(s) with the most ticket sold from 15th September to 15th October 2025.

Provide Relational Algebra expression below:

**1.** **Join the three tables and keep tickets in the date range and confirmed**

Sales = σ\_{issue\_date ≥ 15-09-2025 ∧

issue\_date ≤ 15-10-2025 ∧

status = 'confirmed'}

( AGENT ⋈\_{AGENT.agent\_id = RESERVATION.agent\_id} RESERVATION

⋈\_{RESERVATION.reservation\_id = TICKET.reservation\_id} TICKET )

**2. Count ticket per agent:**

Counts = γ\_{agent\_id, name; tickets\_sold := COUNT(ticket\_id)} (Sales)

**3.Keep only the agent(s) with the maximum no. of tickets:**

Result = σ\_{tickets\_sold = max(tickets\_sold)} (Counts)

Provide SQL query code as TEXT and output below (as image / screenshot / picture):

q1. SQL QUERY

SELECT s.schedule\_id,

r.route\_name,

s.departure\_datetime,

s.ticket\_price,

s.seats\_available,

s.service\_status

FROM schedule s

JOIN route r

ON s.route\_id = r.route\_id

WHERE r.route\_name = 'Newcastle - Berwick Day Tour'

AND s.seats\_available >= 7

AND s.departure\_datetime BETWEEN TRUNC(SYSDATE)

AND TRUNC(SYSDATE) + 14

ORDER BY s.departure\_datetime;

**A computer screen shot of a program

AI-generated content may be incorrect.**

q2. SQL QUERY  
  
SELECT agent\_id,

agent\_name,

tickets\_sold

FROM (

SELECT a.agent\_id,

a.name AS agent\_name,

COUNT(t.ticket\_id) AS tickets\_sold

FROM agent a

JOIN reservation r

ON r.agent\_id = a.agent\_id

JOIN ticket t

ON t.reservation\_id = r.reservation\_id

WHERE t.issue\_date BETWEEN DATE '2025-09-15' AND DATE '2025-10-15'

AND t.status = 'confirmed'

GROUP BY a.agent\_id, a.name

) ticket\_per\_agent

WHERE tickets\_sold = (

SELECT MAX(tickets\_sold)

FROM (

SELECT COUNT(t2.ticket\_id) AS tickets\_sold

FROM agent a2

JOIN reservation r2

ON r2.agent\_id = a2.agent\_id

JOIN ticket t2

ON t2.reservation\_id = r2.reservation\_id

WHERE t2.issue\_date BETWEEN DATE '2025-09-15' AND DATE '2025-10-15'

AND t2.status = 'confirmed'

GROUP BY a2.agent\_id, a2.name

)

)

ORDER BY agent\_name;

A computer screen shot of white text

AI-generated content may be incorrect.

**Part 3 (35 marks)**

This part is based on your answer / solution to Part 1 (A), i.e., conceptual design of the database for the NORTHERNTOURS scenario.

*(A) Choose and justify what aspects of NORTHERNTOURS conceptual design would be better off if implemented using object-relational database; then provide logical design and implementation of the subset of the NORTHERNTOURS using ER/EER to object-relational mapping and object-relational features of Oracle Database System (Kannan, 2019); populate the object-tables with sample data and demonstrate your choice of design and implementation by running two complex queries [e.g., having multiple joins and aggregations] on your object-tables.*

(20 marks)

Answer Part 3 A

1. Provide below your choice and justification of what aspects (subset) of the NORTHERNTOURS conceptual design from Part 1.A you would like to implement using object relational databases (2 marks)

For the object-relational part I chose the route planning subset of my NORTHERNTOURS conceptual design. This subset includes the entities:

**ROUTE**

**ROUTE\_STOP**

**STOP**

In my EER diagram, a route has attributes such as route\_name, total\_distance and duration, and it is associated with an ordered list of stops through the ROUTE\_STOP relationship, which stores stop\_sequence and duration\_min. Each stop is represented by the STOP entity and has attributes such as stop\_name, city, address, parking\_available and facilities.

I picked this subset because it is naturally hierarchical from the business description, one route has many stops in sequence, rather than three completely separate tables. This pattern is well-suited for object-relational features. In Oracle I can model a route as a route object with a collection of stop objects inside it (using object types and nested table collections). This lets me replace the separate ROUTE\_STOP link table for this part of the design and query the route together with all its stops as a single structure, while still keeping STOP as a normal relational table for the rest of the database.

1. Provide below the logical design for your chosen subset using ER/EER to object-relational mapping (2 marks)

**A diagram of a route

AI-generated content may be incorrect.**

ROUTE becomes object type ROUTE\_OBJ with attributes (route\_id, route\_name, description, total\_distance, duration, stops).

ROUTE\_STOP becomes an object type ROUTE\_STOP\_OBJ and a nested table type ROUTE\_STOP\_TAB.

ROUTE\_OR is an object table of ROUTE\_OBJ, with its stops attribute stored as a nested table.

STOP stays as a normal relational table, referenced via stop\_id in ROUTE\_STOP\_OBJ.

1. Provide below the SQL query code as TEXT and output below (as image / screenshot / picture) for implementing your above logical object-relational design (8 marks)

CREATE TYPE ROUTE\_STOP\_OBJ AS OBJECT (

stop\_id NUMBER(10),

stop\_sequence NUMBER(3),

duration\_min NUMBER(5)

);

/

CREATE TYPE ROUTE\_STOP\_TAB AS TABLE OF ROUTE\_STOP\_OBJ;

/

CREATE TYPE ROUTE\_OBJ AS OBJECT (

route\_id NUMBER(10),

route\_name VARCHAR2(100),

description VARCHAR2(200),

total\_distance NUMBER(6,1),

duration NUMBER(5),

stops ROUTE\_STOP\_TAB

);

/

CREATE TABLE ROUTE\_OR OF ROUTE\_OBJ (

CONSTRAINT pk\_route\_or PRIMARY KEY (route\_id)

)

NESTED TABLE stops

STORE AS route\_or\_stops;

**A computer screen shot of a black screen

AI-generated content may be incorrect.**

1. Provide below the SQL query code as TEXT and the output below (as image / screenshot / picture) for populating your above object-relational subset of the NORTHERNTOURS database (4 marks)

New stops for the object-relational routes :

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities)

VALUES ('Newcastle Coach Station',

'Newcastle upon Tyne',

'John Dobson St, Newcastle',

'Y',

'toilets; café; Wi-Fi');

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities)

VALUES ('Morpeth Bus Station',

'Morpeth',

'Bridge St, Morpeth',

'Y',

'toilets; kiosk');

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities)

VALUES ('Alnwick Castle Stop',

'Alnwick',

'Castle Square, Alnwick',

'N',

'photo spot; ticket office nearby');

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities)

VALUES ('Berwick-upon-Tweed Coach Park',

'Berwick-upon-Tweed',

'Ramparts Walk, Berwick',

'Y',

'toilets; café; viewpoint');

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities)

VALUES ('Durham Coach Park',

'Durham',

'Freeman''s Place, Durham',

'Y',

'toilets; café; city centre access');

INSERT INTO stop (stop\_name, city, address, parking\_available, facilities)

VALUES ('York Rail Station',

'York',

'Station Rd, York',

'N',

'toilets; shops; connections');

-- Routes with nested list of stops in ROUTE\_OR.

INSERT INTO route\_or VALUES (

ROUTE\_OBJ(

9001,

'Newcastle - Berwick Coastal Explorer',

'Day tour along the Northumberland coast with photo stops.',

120.0,

420,

ROUTE\_STOP\_TAB(

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'Newcastle Coach Station'),

1,

60

),

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'Morpeth Bus Station'),

2,

60

),

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'Alnwick Castle Stop'),

3,

90

),

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'Berwick-upon-Tweed Coach Park'),

4,

210

)

)

)

);

INSERT INTO route\_or VALUES (

ROUTE\_OBJ(

9002,

'Newcastle - Durham City Highlights',

'Short return trip from Newcastle to Durham with free time in the city.',

50.0,

240,

ROUTE\_STOP\_TAB(

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'Newcastle Coach Station'),

1,

75

),

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'Durham Coach Park'),

2,

165

)

)

)

);

INSERT INTO route\_or VALUES (

ROUTE\_OBJ(

9003,

'Newcastle - York via Durham',

'Full-day tour visiting Durham and York in one trip.',

160.0,

540,

ROUTE\_STOP\_TAB(

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'Newcastle Coach Station'),

1,

60

),

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'Durham Coach Park'),

2,

120

),

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'York Rail Station'),

3,

360

)

)

)

);

INSERT INTO route\_or VALUES (

ROUTE\_OBJ(

9004,

'Durham - Alnwick Castle Day Trip',

'Coach from Durham to Alnwick Castle and back.',

110.0,

420,

ROUTE\_STOP\_TAB(

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'Durham Coach Park'),

1,

90

),

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'Alnwick Castle Stop'),

2,

240

),

ROUTE\_STOP\_OBJ(

(SELECT stop\_id FROM stop WHERE stop\_name = 'Durham Coach Park'),

3,

90

)

)

)

);

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

A screenshot of a computer program

AI-generated content may be incorrect.

1. Provide below the SQL query code as TEXT and output below (as image / screenshot / picture) for running two complex queries on the object-relational subset of the above NORTHERNTOURS database (4 marks).

**Query 1 – Routes with number of stops and total duration:**

SELECT

r.route\_id,

r.route\_name,

COUNT(s.stop\_id) AS number\_of\_stops,

SUM(s.duration\_min) AS total\_stop\_duration\_min

FROM

route\_or r,

TABLE(r.stops) s

GROUP BY

r.route\_id,

r.route\_name

ORDER BY

r.route\_id;  
A screenshot of a computer program

AI-generated content may be incorrect.  
**Query 2 – Routes that pass through a given city (e.g. Durham):**SELECT

r.route\_id,

r.route\_name,

st.city,

COUNT(\*) AS times\_in\_city

FROM

route\_or r,

TABLE(r.stops) s,

stop st

WHERE

s.stop\_id = st.stop\_id

AND st.city = 'Durham'

GROUP BY

r.route\_id,

r.route\_name,

st.city

ORDER BY

times\_in\_city DESC,

r.route\_name;

A screenshot of a computer

AI-generated content may be incorrect.

*(B) Analyse the conceptual database design from Part 1 (A) and the NORTHERNTOURS scenario in the Appendix and propose what aspects of the NORTHERNTOURS database would benefit from incorporating NoSQL Database concepts. Discuss what design choices you made for implementing your chosen subset in NoSQL Database. Illustrate your answer with a representative code from NoSQL Database implementation.*

(15 marks)

Answer Part 3 B

1. Provide below your choice and justification of what aspects (subset) of the NORTHERNTOURS databases would benefit from incorporating NoSQL Database concepts (2 marks)

For the NoSQL element, I decided to analyse the daily sales list per agent in the NORTHERNTOURS system. On paper, an agent has but one list for the day that jumbles together customers (from CUSTOMER), routes (from ROUTE), schedules (from SCHEDULE), tickets (from TICKET) and payments (from TRANSACTION\_). This fits a document authoring style (more so than “strict” tables), since there may be varying number of tickets, payment details and additional notes for each sale. If I go for a document-based database like MongoDB, I can store one JSON doc per agent per day, having a flexible nested structure for the sales, tickets and payments. This has the benefit of not having lots of joins to do and also will be easier to add new fields such as refund which is paid, refund amount or voucher code without needing to redesign the schema.

1. Provide below discussion of what design choices you made for implementing your chosen subset in NoSQL Database (3 marks)

In the NoSQL design I created one main collection called agent\_sales. Each document in this collection shows the sales for one agent on one business day. So the default key is the pair agent + sales\_date. At the top level, the document stores basic fields that come from the AGENT table such as agent\_id, agent name and agent\_city, with that plus sales\_date. The main part is an array called sales, where each element is one booking or sale that the agent made on that day.

In the relational model this data would be divided across the RESERVATION, TICKET, TRANSACTION\_, ROUTE and SCHEDULE tables, but in my NoSQL design I group it together under one roof in the sales array. For example, I save schedule\_id, route\_id, route\_description and departure\_datetime for the trip details. I additionally insert a mini customer sub document containing fields like customer\_id, name and phone, which makes reporting easier and means there are no extra joins to be done. There is a tickets array where each object in the array represents a ticket (ticket\_id, seat, price, status), and a payment sub-document with method, paid\_amount, currency and transaction\_status.

This design is denormalised with respect to the relational schema, but it reflects how the business actually looks at the data: one full sales list per agent per day. It is easy to answer typical questions in a straightforward way, such as “how many tickets did each agent sell in one month?” or “what tickets were cancelled?”, by aggregating over the nested arrays. The structure is not rigidly fixed, if I later want to add optional fields such as promo codes, special notes or several partial payments, no table definition needs to change. This is why a NoSQL document model is useful here.

1. Provide below, code as TEXT and output below (as image / screenshot / picture) for implementing your proposed NoSQL Database subset of the NORTHERNTOURS database, populate it with some data, and example queries & outputs (10 Marks)

db.agent\_sales.insertMany([

{

agent\_id: 201,

agent\_name: 'Tyne Travel Newcastle',

agent\_city: 'Newcastle upon Tyne',

sales\_date: ISODate('2025-09-15T00:00:00Z'),

sales: [

{

sale\_id: ObjectId(),

schedule\_id: 301,

route\_id: 9001,

route\_description: 'Newcastle - Berwick Coastal Explorer',

departure\_datetime: ISODate('2025-09-18T08:30:00Z'),

customer: {

customer\_id: 801,

name: 'Rohan Mehta',

phone: '+44 7700 910111'

},

tickets: [

{ ticket\_id: 91001, seat: '12A', price: 45, status: 'confirmed' },

{ ticket\_id: 91002, seat: '12B', price: 45, status: 'confirmed' }

],

payment: {

method: 'card',

paid\_amount: 90,

currency: 'GBP',

transaction\_status: 'completed'

}

},

{

sale\_id: ObjectId(),

schedule\_id: 302,

route\_id: 9002,

route\_description: 'Newcastle - Durham City Highlights',

departure\_datetime: ISODate('2025-09-19T09:00:00Z'),

customer: {

customer\_id: 802,

name: 'Emily Brown',

phone: '+44 7700 910222'

},

tickets: [

{ ticket\_id: 91003, seat: '5C', price: 30, status: 'confirmed' }

],

payment: {

method: 'cash',

paid\_amount: 30,

currency: 'GBP',

transaction\_status: 'completed'

}

}

]

},

{

agent\_id: 202,

agent\_name: 'Wearside Holiday Agents',

agent\_city: 'Sunderland',

sales\_date: ISODate('2025-09-15T00:00:00Z'),

sales: [

{

sale\_id: ObjectId(),

schedule\_id: 305,

route\_id: 9003,

route\_description: 'Newcastle - York via Durham',

departure\_datetime: ISODate('2025-09-20T07:45:00Z'),

customer: {

customer\_id: 803,

name: 'Michael Carter',

phone: '+1 212 555 0101'

},

tickets: [

{ ticket\_id: 91004, seat: '3A', price: 60, status: 'confirmed' },

{ ticket\_id: 91005, seat: '3B', price: 60, status: 'cancelled' }

],

payment: {

method: 'card',

paid\_amount: 60,

currency: 'GBP',

transaction\_status: 'partially\_refunded'

}

}

]

}

]);

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screen shot of a computer code

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

// Query 1: how many confirmed tickets and total revenue per agent in a date range

db.agent\_sales.aggregate([

{

$match: {

sales\_date: {

$gte: ISODate('2025-09-01T00:00:00Z'),

$lte: ISODate('2025-10-01T00:00:00Z')

}

}

},

{ $unwind: '$sales' },

{ $unwind: '$sales.tickets' },{

$match: {

'sales.tickets.status': 'confirmed'

}

},

{

$group: {

\_id: { agent\_id: '$agent\_id', agent\_name: '$agent\_name' },

tickets\_sold: { $sum: 1 },

total\_revenue: { $sum: '$sales.tickets.price' }

}

},

{ $sort: { tickets\_sold: -1, '\_id.agent\_name': 1 } }

]);

**A screen shot of a computer program

AI-generated content may be incorrect.**

// Query 2: find sales for a route where any ticket was cancelled (e.g. route 9003)

db.agent\_sales.find(

{

'sales.route\_id': 9003,

'sales.tickets.status': 'cancelled'

},

{

agent\_name: 1,

agent\_city: 1,

sales\_date: 1,

'sales.$': 1

}

);

A screen shot of a computer program

AI-generated content may be incorrect.

**Part 4 (10 marks)**

*Consider the NORTHERNTOURS scenario in the Appendix. Produce a report for the managing director of the NORTHERNTOURS company – elaborating on sustainability, professional, legal, ethical and security issues, and matters related to diversity, inclusion, cultural, societal and environment as well as risk management and evaluation of commercial risks that need to be considered and make appropriate recommendations for NORTHERNTOURS. Whilst answering this question, employ a critical review of current and relevant literature, systems, developments, and standards.*

(10 marks)

The report should be concise and comprehensive and in the region of 900-1000 words. You should use Harvard style of citation and referencing by following the guidelines in Pears and Shields (2008).

**Answer Part 4: 10 Marks** [8 for the quality of report covering all of the above issues, 1 for the quality of referencing and citation and adhering to the Harvard style, 1 for presentation]

**Introduction**

This report outlines the factors that NORTHERNTOURS should consider when creating a new database system. The new system will improve customer service and make work faster. There are problem which need to be solved in the right way that is business risks, cultural differences, laws and rules, access for disabled people, data security, and environmental sustainability. These are the some important issues which we must also handle. I will explain each point in simple language and will also tell you what practical advice should be adopted.

**Sustainability For The Environment**

The coach industry in the UK is currently undergoing significant changes. In 2024, the market reached a 16-year record, with 8,390 new coaches entering the road a full 70% increase from the previous year (SMMT, 2025). Of these, 1,570 will be electric or hydrogen-powered, making the UK the largest green bus market in Europe ahead of Italy, Germany, and France. This means that environmental action is no longer just talk, but is being implemented in practice. Our database can also help NORTHERNTOURS achieve similar environmental goals. The COACH table allows us to determine which older diesel trains should be replaced with electric ones by looking at the age and service date of the trains. The ROUTE and SCHEDULE tables allow routes to be set to minimize unnecessary trips, reducing fuel consumption and pollution. The adoption of digital tickets will also significantly reduce paper waste. Nowadays, many customers prefer environmentally friendly companies, so these things can also increase bookings. I recommend tracking carbon emissions for each route, setting annual reduction targets, and using renewable-energy cloud servers instead of physical servers.

**Privacy and Data Security**

We have to protect customer data because we deal with personal information including names, addresses, phone numbers, emails, and credit card numbers. The UK GDPR has very stringent regulations. In 2024, there were significant fines. The Police Service of Northern Ireland received £750,000 for a security breach that the Information Commissioner's Office termed as the greatest data breach in UK policing history, and LinkedIn was fined €310 million for exploiting client information (Skillcast, 2024). The government's serious approach to data protection is demonstrated by these harsh sanctions. The TRANSACTION\_ table stores payment card information requiring extra caution. All credit card data must be encrypted per PCI DSS security guidelines so hackers cannot read it if they breach the system. We need role-based access controls - drivers should not see payment details, clerks should not access driver information. Only the data required for their jobs is visible to each individual. Hacking of data must be reported to the ICO within 72 hours according to UK law, even on weekends . There is detailed staff training on how to secure data and written plans for how to respond to incidents are all important precautions to take that is Regular security testing, strong passwords, two-factor authentication that requires a password and a phone code.

**Ethics and Legal Requirements**

Many laws affect coach operations. Consumer Rights Act 2015 entitles customers to refunds for cancellations or delays, requiring clear policies and easy refund processing through TRANSACTION\_ table. Equality Act 2010 requires helping disabled passengers, this is legal requirement, not optional, and we can be prosecuted for non-compliance. We must record which coaches have wheelchair access and which customers requested assistance. To cut down on accidents caused by tired drivers, EU Regulation 561/2006 limits working hours (European Parliament, 2006). SCHEDULE and DRIVER tables should check compliance automatically. We shouldn't share client contact information with other businesses without their authorisation. We need to be open about how we utilise data and only gather data that we really need.

**Inclusion, Diversity and Accessibility**

The law says indicate the names of routes and the stops that are coming up should have sensory displays in all the buses (UK Government, 2024). Starting in October 2024 This significantly helps passengers with vision or hearing difficulties. Bus Users UK research found 70% of visually impaired passengers miss stops because drivers forget announcements. The government provided £4.65 million helping smaller companies install announcement systems. Our database should track which coaches have these features and which need upgrading. STOP\_MEAL table has vegetarian options but should expand to vegan, halal, kosher meals and allergy information for religious, health, or personal dietary needs. Adding language preferences to CUSTOMER table helps tourists and non-english speakers. Culturally sensitive route planning to religious sites, festivals, or heritage locations broadens customer reach while demonstrating community values.

**Management of Business Risks**

Several business risks need proper management. First, overbooking selling more tickets than seats causes customer anger and legal issues. SEATS\_AVAILABLE field prevents this but needs automatic low seat alerts. Second, system failure stops bookings and ticket checks. We need hourly backups and tested backup systems taking over if main system fails. Test twice yearly. Third, competition from trains, airlines, car rental, and Uber requires competitive prices and excellent service. Dynamic pricing adjusts prices with demand. The 2024 Buses Bill may change English bus operations. Fourth, payment fraud using stolen cards needs detection systems. Fifth, seasonal variations busy summers, quiet winters require smart capacity planning. Finally, post-Brexit driver shortages make recruitment difficult. The database should track driver satisfaction to address problems before losing staff.

Conclusion and Suggestions

Implemented properly with these considerations, the database system will be valuable for NORTHERNTOURS. Most urgent priorities are:

• Appointing someone responsible for GDPR compliance

• Encrypting all payment card data properly

• Setting up hourly backups with tested disaster recovery

• Adding accessibility and dietary information fields

We must develop a realistic plan replacing older diesel coaches with electric vehicles over coming years. All employees need training on data security regulations and assisting disabled passengers. By following legal requirements, securing customer information, serving disabled passengers properly, minimizing environmental impact, and managing business risks wisely, the database becomes more than a computer system. It becomes a strategic tool helping NORTHERNTOURS grow successfully as a responsible and trustworthy company customers choose.

**References & Bibliography**

**Bus Users UK (2024)** *Why are we waiting? Disabled people's experiences of travelling by bus*. Available at: <https://bususers.org/uk/new-report-exposes-accessibility-challenges-for-disabled-bus-passengers/> (Accessed: 3 December 2025). 

**European Parliament (2006)** Regulation (EC) No 561/2006 on driving times and rest periods. Brussels: European Union. Available at: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32006R0561> 

**Floridi, L. and Taddeo, M. (2016)** 'What is data ethics?', *Philosophical Transactions of the Royal Society A*, 374(2083), pp. 1-5. Available at: <https://royalsocietypublishing.org/doi/10.1098/rsta.2016.0360> 

**Skillcast (2024)** '20 biggest GDPR fines 2018-2024', *Compliance Review*. Available at: <https://www.skillcast.com/blog/20-biggest-gdpr-fines> (Accessed: 3 December 2025).

**SMMT (2025)** 'Britain's bus market soars to 16 year high', *Society of Motor Manufacturers and Traders Press Release*, 15 February. Available at: <https://www.smmt.co.uk/britains-bus-market-soars-to-16-year-high-and-remains-europes-biggest-zero-emission-buyer/> (Accessed: 3 December 2025). 

**Sustainable Bus (2024)** 'Is the Buses Bill going to shake the UK bus industry up?', 29 November. Available at: <https://www.sustainable-bus.com/news/uk-buses-bill-franchise-public-transport/> (Accessed: 3 December 2025). 

**UK Government (2024)** 'New regulations come into force to improve accessibility on local buses and coaches', *Department for Transport*, 1 October. Available at: <https://www.gov.uk/government/news/new-regulations-come-into-force-to-improve-accessibility-on-local-buses-and-coaches> (Accessed: 3 December 2025). 

**UK Parliament (2010)** *Equality Act 2010*. London: The Stationery Office. Available at: <https://www.legislation.gov.uk/ukpga/2010/15/contents> 

**UK Parliament (2015)** *Consumer Rights Act 2015*. London: The Stationery Office.  
Available at: <https://www.legislation.gov.uk/ukpga/2015/15/contents> 

**UK Parliament (2018)** *Data Protection Act 2018*. London: The Stationery Office. Available at: <https://www.legislation.gov.uk/ukpga/2018/12/contents>