KL7011_w24041293.docx

by Sumit MALVIYA

Submission date: 04-Dec-2025 03:49PM (UTC+0000)
Submission ID: 269609525
File name: KL7011_w24041293.docx (5.1M)
Word count: 9353
Character count: 60804

Northumbria University NEWCASTLE

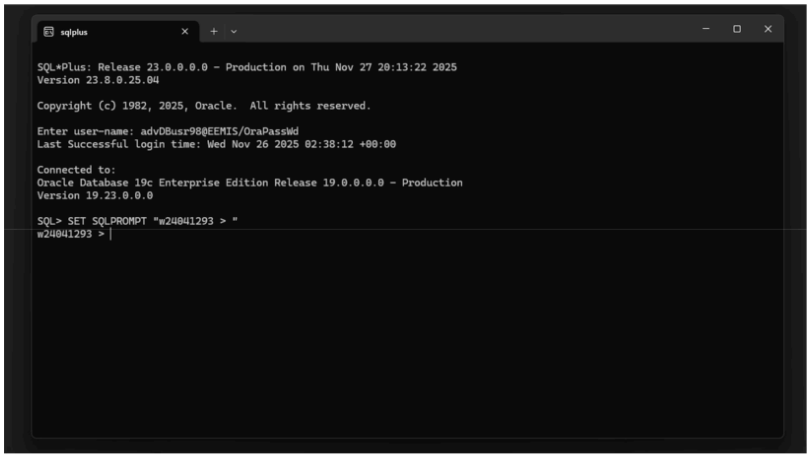
Department of Computer & Information Sciences

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
% 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:



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

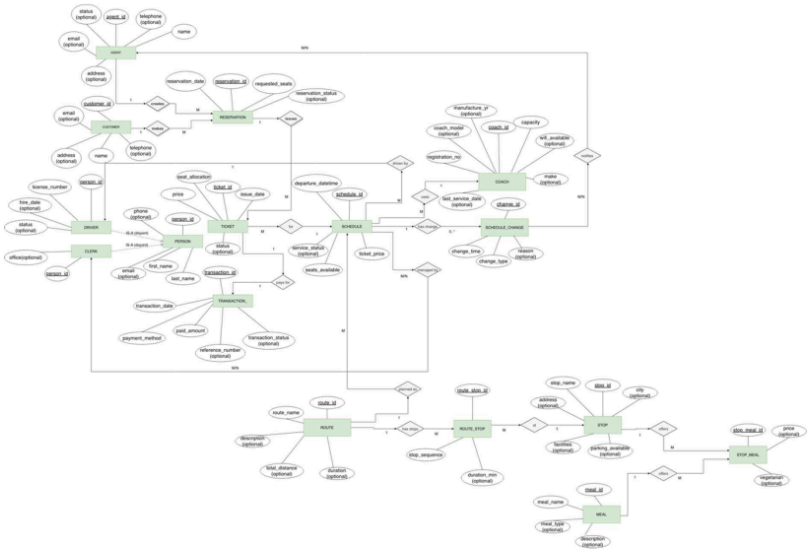
Part 1 (40 marks)

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

(A) 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



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.

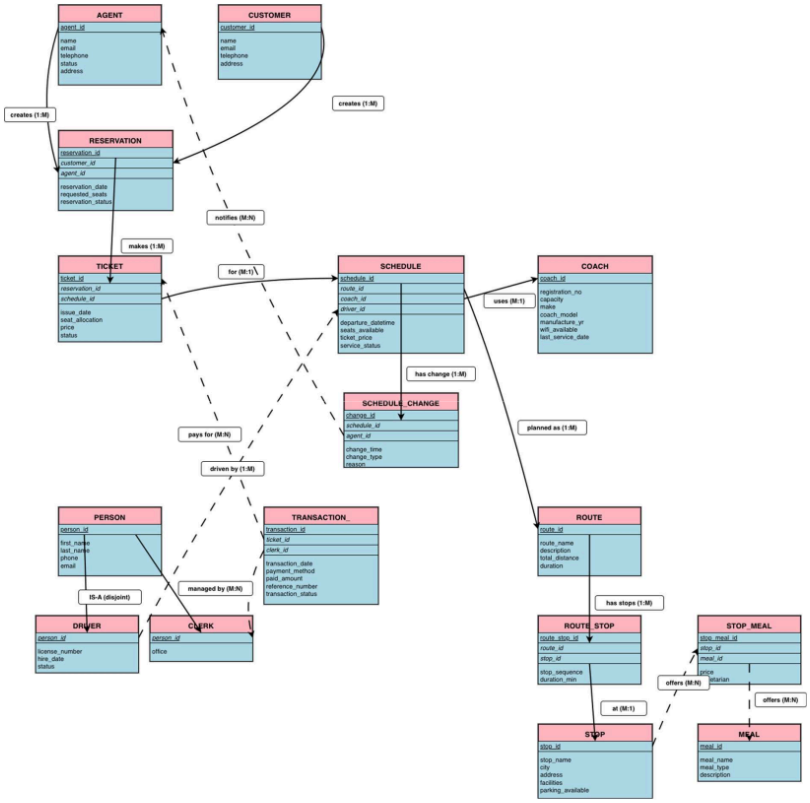
My EER diagram is designed to match the main needs of the NorthernTours coach booking scenario. The PERSON–DRIVER–CLERK structure lets me keep common PERSON details in one place, while still showing that DRIVERS and CLERKS have different roles and attributes. AGENT and CUSTOMER are linked to reservation,

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.

(B) 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)



- **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:

SCHEDULE.ROUTE_ID → ROUTE.ROUTE_ID
SCHEDULE.COACH_ID → COACH.COACH_ID
SCHEDULE.DRIVER_ID → DRIVER.PERSON_ID
TICKET.SCHEDULE_ID → SCHEDULE.SCHEDULE_ID
TICKET.RESERVATION_ID → RESERVATION.RESERVATION_ID
RESERVATION.AGENT_ID → AGENT.AGENT_ID
RESERVATION.CUSTOMER_ID → CUSTOMER.CUSTOMER_ID
SCHEDULE_CHANGE.SCHEDULE_ID → SCHEDULE.SCHEDULE_ID

TRANSACTION_.TICKET_ID → TICKET.TICKET_ID

- **M:N relationships → junction tables** M:N relationships are mapped to separate tables with composite keys:

ROUTE_STOP(ROUTE_STOP_ID, ROUTE_ID, STOP_SEQUENCE, STOP_ID, DURATION_MIN)
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:

Primary key: (ROUTE_ID, STOP_SEQUENCE) keeps the sequence of stops.
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:

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:

PERSON_ID is both the primary key and a foreign key to PERSON(PERSON_ID).
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.

(C) 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),
   licens_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.

```
w24041293 >
w24041293 > CREATE TABLE PERSON (
  2      person_id   NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      first_name  VARCHAR2(50)  NOT NULL,
  4      last_name   VARCHAR2(50)  NOT NULL,
  5      phone       VARCHAR2(30),
  6      email       VARCHAR2(120)
  7  );

Table created.

w24041293 >
w24041293 > CREATE TABLE DRIVER (
  2      person_id       NUMBER(10),
  3      license_number  VARCHAR2(30) NOT NULL,
  4      hire_date       DATE,
  5      status          VARCHAR2(20)
  6  );

Table created.

w24041293 >
w24041293 > CREATE TABLE CLERK (
  2      person_id  NUMBER(10),
  3      office     VARCHAR2(50)
  4  );

Table created.

w24041293 >
w24041293 > CREATE TABLE AGENT (
  2      agent_id   NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      name       VARCHAR2(100) NOT NULL,
  4      email      VARCHAR2(120),
  5      telephone  VARCHAR2(30),
  6      status     VARCHAR2(20),
  7      address    VARCHAR2(200)
  8  );

Table created.
```

```
w24041293 >
w24041293 > CREATE TABLE CUSTOMER (
  2      customer_id  NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      name         VARCHAR2(100) NOT NULL,
  4      email        VARCHAR2(120),
  5      telephone    VARCHAR2(30),
  6      address      VARCHAR2(200)
  7  );

Table created.

w24041293 >
w24041293 > CREATE TABLE COACH (
  2      coach_id          NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      registration_no   VARCHAR2(20) NOT NULL,
  4      capacity          NUMBER(3)    NOT NULL,
  5      make              VARCHAR2(50),
  6      manufacture_yr    NUMBER(4),
  7      last_service_date DATE,
  8      wifi_available    CHAR(1),
  9      coach_model       VARCHAR2(50)
 10  );

Table created.

w24041293 >
w24041293 > CREATE TABLE ROUTE (
  2      route_id       NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      route_name     VARCHAR2(100) NOT NULL,
  4      description    VARCHAR2(200),
  5      total_distance NUMBER(6,1),
  6      duration       NUMBER(5)
  7  );

Table created.

w24041293 >
w24041293 > CREATE TABLE STOP (
  2      stop_id           NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      stop_name         VARCHAR2(100) NOT NULL,
  4      city              VARCHAR2(100),
  5      address           VARCHAR2(200),
  6      parking_available CHAR(1),
  7      facilities        VARCHAR2(200)
  8  );

Table created.
```

```
w24041293 >
w24041293 > CREATE TABLE MEAL (
  2      meal_id     NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      meal_name   VARCHAR2(100) NOT NULL,
  4      meal_type   VARCHAR2(50),
  5      description VARCHAR2(200)
  6  );

Table created.

w24041293 >
w24041293 > CREATE TABLE ROUTE_STOP (
  2      route_id       NUMBER(10) NOT NULL,
  3      stop_sequence  NUMBER(3)  NOT NULL,
  4      stop_id        NUMBER(10) NOT NULL,
  5      duration_min   NUMBER(5)
  6  );

Table created.

w24041293 >
w24041293 > CREATE TABLE STOP_MEAL (
  2      stop_id     NUMBER(10) NOT NULL,
  3      meal_id     NUMBER(10) NOT NULL,
  4      price       NUMBER(6,2),
  5      vegetarian  CHAR(1)
  6  );

Table created.

w24041293 >
w24041293 > CREATE TABLE SCHEDULE (
  2      schedule_id        NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      route_id           NUMBER(10) NOT NULL,
  4      coach_id           NUMBER(10) NOT NULL,
  5      driver_id          NUMBER(10) NOT NULL,
  6      departure_datetime DATE       NOT NULL,
  7      ticket_price       NUMBER(6,2) NOT NULL,
  8      seats_available    NUMBER(3),
  9      service_status     VARCHAR2(20)
 10  );

Table created.
```

```
w24041293 >
w24041293 > CREATE TABLE SCHEDULE_CHANGE (
  2      change_id    NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      schedule_id  NUMBER(10) NOT NULL,
  4      change_time  DATE       NOT NULL,
  5      change_type  VARCHAR2(20) NOT NULL,
  6      reason       VARCHAR2(200)
  7  );

Table created.

w24041293 >
w24041293 > CREATE TABLE RESERVATION (
  2      reservation_id     NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      agent_id           NUMBER(10) NOT NULL,
  4      customer_id        NUMBER(10) NOT NULL,
  5      reservation_date   DATE       NOT NULL,
  6      requested_seats    NUMBER(3)  NOT NULL,
  7      reservation_status VARCHAR2(20)
  8  );

Table created.

w24041293 >
w24041293 > CREATE TABLE TICKET (
  2      ticket_id       NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      reservation_id  NUMBER(10) NOT NULL,
  4      schedule_id     NUMBER(10) NOT NULL,
  5      issue_date      DATE       NOT NULL,
  6      seat_allocation VARCHAR2(10) NOT NULL,
  7      price           NUMBER(6,2) NOT NULL,
  8      status          VARCHAR2(20)
  9  );

Table created.

w24041293 >
w24041293 > CREATE TABLE TRANSACTION_ (
  2      transaction_id     NUMBER(10) GENERATED ALWAYS AS IDENTITY,
  3      ticket_id          NUMBER(10) NOT NULL,
  4      transaction_date   DATE       NOT NULL,
  5      payment_method     VARCHAR2(20) NOT NULL,
  6      paid_amount        NUMBER(8,2)  NOT NULL,
  7      reference_number   VARCHAR2(30),
  8      transaction_status VARCHAR2(20)
  9  );

Table created.
```

SQL DDL Constraints (ALTER TABLE statements):

```
w24041293 > ALTER TABLE PERSON
  2    ADD CONSTRAINT pk_person PRIMARY KEY (person_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE DRIVER
  2    ADD CONSTRAINT pk_driver PRIMARY KEY (person_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE CLERK
  2    ADD CONSTRAINT pk_clerk PRIMARY KEY (person_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE AGENT
  2    ADD CONSTRAINT pk_agent PRIMARY KEY (agent_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE CUSTOMER
  2    ADD CONSTRAINT pk_customer PRIMARY KEY (customer_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE COACH
  2    ADD CONSTRAINT pk_coach PRIMARY KEY (coach_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE ROUTE
  2    ADD CONSTRAINT pk_route PRIMARY KEY (route_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE STOP
  2    ADD CONSTRAINT pk_stop PRIMARY KEY (stop_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE MEAL
  2    ADD CONSTRAINT pk_meal PRIMARY KEY (meal_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE ROUTE_STOP
  2    ADD CONSTRAINT pk_route_stop PRIMARY KEY (route_id, stop_sequence);

Table altered.
```

```
w24041293 >
w24041293 > ALTER TABLE STOP_MEAL
  2    ADD CONSTRAINT pk_stop_meal PRIMARY KEY (stop_id, meal_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE SCHEDULE
  2    ADD CONSTRAINT pk_schedule PRIMARY KEY (schedule_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE SCHEDULE_CHANGE
  2    ADD CONSTRAINT pk_schedule_change PRIMARY KEY (change_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE RESERVATION
  2    ADD CONSTRAINT pk_reservation PRIMARY KEY (reservation_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE TICKET
  2    ADD CONSTRAINT pk_ticket PRIMARY KEY (ticket_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE TRANSACTION_
  2    ADD CONSTRAINT pk_transaction PRIMARY KEY (transaction_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE DRIVER
  2    ADD CONSTRAINT fk_driver_person
  3        FOREIGN KEY (person_id)
  4        REFERENCES PERSON(person_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE CLERK
  2    ADD CONSTRAINT fk_clerk_person
  3        FOREIGN KEY (person_id)
  4        REFERENCES PERSON(person_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE ROUTE_STOP
  2    ADD CONSTRAINT fk_route_stop_route
  3        FOREIGN KEY (route_id)
  4        REFERENCES ROUTE(route_id);

Table altered.
```

```
w24041293 >
w24041293 > ALTER TABLE ROUTE_STOP
  2    ADD CONSTRAINT fk_route_stop_stop
  3        FOREIGN KEY (stop_id)
  4        REFERENCES STOP(stop_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE STOP_MEAL
  2    ADD CONSTRAINT fk_stop_meal_stop
  3        FOREIGN KEY (stop_id)
  4        REFERENCES STOP(stop_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE STOP_MEAL
  2    ADD CONSTRAINT fk_stop_meal_meal
  3        FOREIGN KEY (meal_id)
  4        REFERENCES MEAL(meal_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE SCHEDULE
  2    ADD CONSTRAINT fk_schedule_route
  3        FOREIGN KEY (route_id)
  4        REFERENCES ROUTE(route_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE SCHEDULE
  2    ADD CONSTRAINT fk_schedule_coach
  3        FOREIGN KEY (coach_id)
  4        REFERENCES COACH(coach_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE SCHEDULE
  2    ADD CONSTRAINT fk_schedule_driver
  3        FOREIGN KEY (driver_id)
  4        REFERENCES DRIVER(person_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE SCHEDULE_CHANGE
  2    ADD CONSTRAINT fk_schedule_change_schedule
  3        FOREIGN KEY (schedule_id)
  4        REFERENCES SCHEDULE(schedule_id);

Table altered.
```

```
w24041293 >
w24041293 > ALTER TABLE RESERVATION
  2    ADD CONSTRAINT fk_reservation_agent
  3        FOREIGN KEY (agent_id)
  4        REFERENCES AGENT(agent_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE RESERVATION
  2    ADD CONSTRAINT fk_reservation_customer
  3        FOREIGN KEY (customer_id)
  4        REFERENCES CUSTOMER(customer_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE TICKET
  2    ADD CONSTRAINT fk_ticket_reservation
  3        FOREIGN KEY (reservation_id)
  4        REFERENCES RESERVATION(reservation_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE TICKET
  2    ADD CONSTRAINT fk_ticket_schedule
  3        FOREIGN KEY (schedule_id)
  4        REFERENCES SCHEDULE(schedule_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE TRANSACTION_
  2    ADD CONSTRAINT fk_transaction_ticket
  3        FOREIGN KEY (ticket_id)
  4        REFERENCES TICKET(ticket_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE DRIVER
  2    ADD CONSTRAINT uq_driver_license UNIQUE (license_number);

Table altered.

w24041293 >
w24041293 > ALTER TABLE COACH
  2    ADD CONSTRAINT uq_coach_reg UNIQUE (registration_no);

Table altered.

w24041293 >
w24041293 > ALTER TABLE ROUTE
  2    ADD CONSTRAINT uq_route_name UNIQUE (route_name);

Table altered.
```

```
w24041293 >
w24041293 > ALTER TABLE ROUTE_STOP
  2    ADD CONSTRAINT uq_route_stop UNIQUE (route_id, stop_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE TRANSACTION_
  2    ADD CONSTRAINT uq_transaction_ticket UNIQUE (ticket_id);

Table altered.

w24041293 >
w24041293 > ALTER TABLE COACH
  2    ADD CONSTRAINT ck_coach_capacity
  3        CHECK (capacity > 0);

Table altered.

w24041293 >
w24041293 > ALTER TABLE COACH
  2    ADD CONSTRAINT ck_coach_wifi
  3        CHECK (wifi_available IN ('Y','N') OR wifi_available IS NULL);

Table altered.

w24041293 >
w24041293 > ALTER TABLE STOP
  2    ADD CONSTRAINT ck_stop_parking
  3        CHECK (parking_available IN ('Y','N') OR parking_available IS NULL);

Table altered.

w24041293 >
w24041293 > ALTER TABLE STOP_MEAL
  2    ADD CONSTRAINT ck_stop_meal_veg
  3        CHECK (vegetarian IN ('Y','N') OR vegetarian IS NULL);

Table altered.

w24041293 >
w24041293 > ALTER TABLE SCHEDULE
  2    ADD CONSTRAINT ck_schedule_seats
  3        CHECK (seats_available >= 0 OR seats_available IS NULL);

Table altered.

w24041293 >
w24041293 > ALTER TABLE SCHEDULE
  2    ADD CONSTRAINT ck_schedule_status
  3        CHECK (
  4          service_status IN ('scheduled','delayed','cancelled','completed')
  5          OR service_status IS NULL
  6        );

Table altered.
```

```
w24041293 >
w24041293 > ALTER TABLE RESERVATION
  2    ADD CONSTRAINT ck_reservation_seats
  3        CHECK (requested_seats > 0);

Table altered.

w24041293 >
w24041293 > ALTER TABLE TRANSACTION_
  2    ADD CONSTRAINT ck_transaction_amount
  3        CHECK (paid_amount >= 0);

Table altered.

w24041293 >
```

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

```
w24041293 > @E:\northerntours\scripts\northerntours_sample_data_subset.sql
w24041293 > INSERT INTO person (first_name, last_name, phone, email) VALUES
  2    ('Aditi',   'Sharma',  '+44 7700 900111', 'aditi.sharma@northerntours.co.uk');

1 row created.

w24041293 > INSERT INTO person (first_name, last_name, phone, email) VALUES
  2    ('Emily',   'Watson',  '+44 7700 900222', 'emily.watson@northerntours.co.uk');

1 row created.

w24041293 > INSERT INTO person (first_name, last_name, phone, email) VALUES
  2    ('Michael', 'Carter',  '+1 212 555 0101', 'michael.carter@northerntours.com');

1 row created.

w24041293 > INSERT INTO person (first_name, last_name, phone, email) VALUES
  2    ('Rohan',   'Patel',   '+91 98 7654 3210','rohan.patel@northerntours.co.uk');

1 row created.

w24041293 > INSERT INTO person (first_name, last_name, phone, email) VALUES
  2    ('Sarah',   'Johnson', '+1 415 555 0202', 'sarah.johnson@northerntours.com');

1 row created.

w24041293 > INSERT INTO person (first_name, last_name, phone, email) VALUES
  2    ('Liam',    'Brown',   '+44 7700 900333', 'liam.brown@northerntours.co.uk');

1 row created.

w24041293 > INSERT INTO person (first_name, last_name, phone, email) VALUES
  2    ('Olivia',  'Davis',   '+44 7700 900444', 'olivia.davis@northerntours.co.uk');

1 row created.

w24041293 > INSERT INTO person (first_name, last_name, phone, email) VALUES
  2    ('Noah',    'Wilson',  '+44 7700 900555', 'noah.wilson@northerntours.co.uk');

1 row created.

w24041293 > INSERT INTO person (first_name, last_name, phone, email) VALUES
  2    ('Zara',    'Khan',    '+44 7700 900666', 'zara.khan@northerntours.co.uk');

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO driver (person_id, license_number, hire_date, status)
  2  SELECT person_id, 'UK-DRV-1001', TO_DATE('2020-03-15','YYYY-MM-DD'), 'active'
  3  FROM person
  4  WHERE first_name = 'Aditi' AND last_name = 'Sharma';

1 row created.

w24041293 >
w24041293 > INSERT INTO driver (person_id, license_number, hire_date, status)
  2  SELECT person_id, 'UK-DRV-1002', TO_DATE('2019-06-01','YYYY-MM-DD'), 'active'
  3  FROM person
  4  WHERE first_name = 'Emily' AND last_name = 'Watson';

1 row created.

w24041293 >
w24041293 > INSERT INTO driver (person_id, license_number, hire_date, status)
  2  SELECT person_id, 'US-DRV-2001', TO_DATE('2018-09-10','YYYY-MM-DD'), 'active'
  3  FROM person
  4  WHERE first_name = 'Michael' AND last_name = 'Carter';

1 row created.

w24041293 >
w24041293 > INSERT INTO driver (person_id, license_number, hire_date, status)
  2  SELECT person_id, 'UK-DRV-3001', TO_DATE('2021-01-05','YYYY-MM-DD'), 'active'
  3  FROM person
  4  WHERE first_name = 'Liam' AND last_name = 'Brown';

1 row created.

w24041293 >
w24041293 > INSERT INTO driver (person_id, license_number, hire_date, status)
  2  SELECT person_id, 'UK-DRV-4001', TO_DATE('2022-02-10','YYYY-MM-DD'), 'active'
  3  FROM person
  4  WHERE first_name = 'Noah' AND last_name = 'Wilson';

1 row created.

w24041293 >
w24041293 > INSERT INTO driver (person_id, license_number, hire_date, status)
  2  SELECT person_id, 'UK-DRV-5001', TO_DATE('2023-04-18','YYYY-MM-DD'), 'active'
  3  FROM person
  4  WHERE first_name = 'Zara' AND last_name = 'Khan';

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO agent (name, email, telephone, status, address) VALUES
  2    ('Northern Travels Newcastle', 'info@northerntravels.co.uk',
  3     '+44 191 555 0100', 'active', 'Newcastle upon Tyne, UK');

1 row created.

w24041293 > INSERT INTO agent (name, email, telephone, status, address) VALUES
  2    ('Durham City Breaks', 'sales@durhambreaks.co.uk',
  3     '+44 191 555 0200', 'active', 'Durham, UK');

1 row created.

w24041293 > INSERT INTO agent (name, email, telephone, status, address) VALUES
  2    ('Global Trips London', 'contact@globaltrips.co.uk',
  3     '+44 20 7000 1234', 'active', 'London, UK');

1 row created.

w24041293 > INSERT INTO agent (name, email, telephone, status, address) VALUES
  2    ('Sunshine Tours Mumbai', 'hello@sunshinetours.in',
  3     '+91 22 4000 5678', 'active', 'Mumbai, India');

1 row created.

w24041293 > INSERT INTO agent (name, email, telephone, status, address) VALUES
  2    ('Pacific Getaways New York', 'bookings@pacificgetaways.com',
  3     '+1 212 555 0909', 'active', 'New York, USA');

1 row created.

w24041293 > INSERT INTO agent (name, email, telephone, status, address) VALUES
  2    ('Highland Explorer Tours', 'info@highlandexplorer.co.uk',
  3     '+44 131 555 0300', 'active', 'Edinburgh, UK');

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO customer (name, email, telephone, address) VALUES
  2    ('Priya Singh', 'priya.singh@example.com', '+44 7700 111111',
  3     'Gosforth, Newcastle upon Tyne, UK');

1 row created.

w24041293 > INSERT INTO customer (name, email, telephone, address) VALUES
  2    ('Oliver Green', 'oliver.green@example.com', '+44 7700 222222',
  3     'Jesmond, Newcastle upon Tyne, UK');

1 row created.

w24041293 > INSERT INTO customer (name, email, telephone, address) VALUES
  2    ('Sophia Miller', 'sophia.miller@example.com', '+1 415 555 0303',
  3     'San Francisco, USA');

1 row created.

w24041293 > INSERT INTO customer (name, email, telephone, address) VALUES
  2    ('Arjun Rao', 'arjun.rao@example.com', '+91 98 1111 2222',
  3     'Bangalore, India');

1 row created.

w24041293 > INSERT INTO customer (name, email, telephone, address) VALUES
  2    ('Chloe Anderson', 'chloe.anderson@example.com', '+44 7700 333333',
  3     'Durham, UK');

1 row created.

w24041293 > INSERT INTO customer (name, email, telephone, address) VALUES
  2    ('Ethan Walker', 'ethan.walker@example.com', '+44 7700 444444',
  3     'York, UK');

1 row created.

w24041293 > INSERT INTO customer (name, email, telephone, address) VALUES
  2    ('Mia Patel', 'mia.patel@example.com', '+44 7700 555555',
  3     'Sunderland, UK');

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO coach (registration_no, capacity, make, manufacture_yr,
  2                    last_service_date, wifi_available, coach_model)
  3  VALUES ('NT01 NCL', 50, 'Volvo', 2020,
  4          TO_DATE('2024-06-01','YYYY-MM-DD'), 'Y', 'Volvo 9700');

1 row created.

w24041293 > INSERT INTO coach (registration_no, capacity, make, manufacture_yr,
  2                    last_service_date, wifi_available, coach_model)
  3  VALUES ('NT02 DUR', 40, 'Mercedes', 2019,
  4          TO_DATE('2024-05-15','YYYY-MM-DD'), 'Y', 'Tourismo');

1 row created.

w24041293 > INSERT INTO coach (registration_no, capacity, make, manufacture_yr,
  2                    last_service_date, wifi_available, coach_model)
  3  VALUES ('NT03 EDI', 55, 'Scania', 2018,
  4          TO_DATE('2024-04-20','YYYY-MM-DD'), 'N', 'Irizar i6');

1 row created.

w24041293 > INSERT INTO coach (registration_no, capacity, make, manufacture_yr,
  2                    last_service_date, wifi_available, coach_model)
  3  VALUES ('NT04 YORK', 45, 'MAN', 2021,
  4          TO_DATE('2024-07-10','YYYY-MM-DD'), 'Y', 'Lion Coach');

1 row created.

w24041293 > INSERT INTO coach (registration_no, capacity, make, manufacture_yr,
  2                    last_service_date, wifi_available, coach_model)
  3  VALUES ('NT05 NCL2', 48, 'Volvo', 2022,
  4          TO_DATE('2024-07-20','YYYY-MM-DD'), 'Y', 'Volvo 9900');

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO route (route_name, description, total_distance, duration) VALUES
  2    ('Newcastle - Berwick Day Tour',
  3     'Scenic coastal route from Newcastle to Berwick-upon-Tweed',
  4     90.0, 120);

1 row created.

w24041293 > INSERT INTO route (route_name, description, total_distance, duration) VALUES
  2    ('Newcastle - Durham Commuter',
  3     'Quick commuter route between Newcastle and Durham',
  4     30.0, 45);

1 row created.

w24041293 > INSERT INTO route (route_name, description, total_distance, duration) VALUES
  2    ('Edinburgh - Berwick Coastal',
  3     'Coastal service from Edinburgh to Berwick-upon-Tweed',
  4     60.0, 90);

1 row created.

w24041293 > INSERT INTO route (route_name, description, total_distance, duration) VALUES
  2    ('York - Newcastle Express',
  3     'Express coach from York to Newcastle',
  4     130.0, 150);

1 row created.

w24041293 > INSERT INTO route (route_name, description, total_distance, duration) VALUES
  2    ('Newcastle - Edinburgh Direct',
  3     'Direct service from Newcastle to Edinburgh',
  4     160.0, 180);

1 row created.

w24041293 > INSERT INTO route (route_name, description, total_distance, duration) VALUES
  2    ('Durham - York Shuttle',
  3     'Shuttle service between Durham and York',
  4     95.0, 110);

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities) VALUES
  2    ('Newcastle Coach Station', 'Newcastle upon Tyne',
  3     'Neville Street, Newcastle upon Tyne', 'Y',
  4     'Toilets; Cafe; Waiting area');

1 row created.

w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities) VALUES
  2    ('Berwick-upon-Tweed Bus Station', 'Berwick-upon-Tweed',
  3     'Walkergate, Berwick-upon-Tweed', 'N',
  4     'Sheltered stands');

1 row created.

w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities) VALUES
  2    ('Durham Bus Station', 'Durham',
  3     'North Road, Durham', 'Y',
  4     'Shops; Toilets');

1 row created.

w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities) VALUES
  2    ('Morpeth Bus Station', 'Morpeth',
  3     'Bridge Street, Morpeth', 'N',
  4     'Cafe; Seating');

1 row created.

w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities) VALUES
  2    ('Edinburgh Bus Station', 'Edinburgh',
  3     'Elder Street, Edinburgh', 'Y',
  4     'Shops; Toilets; WiFi');

1 row created.

w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities) VALUES
  2    ('York Coach Station', 'York',
  3     'Station Road, York', 'Y',
  4     'Cafe; Toilets; Waiting room');

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO route_stop (route_id, stop_sequence, stop_id, duration_min)
  2  VALUES (
  3    (SELECT route_id FROM route
  4     WHERE route_name = 'Newcastle - Berwick Day Tour'),
  5    1,
  6    (SELECT stop_id FROM stop
  7     WHERE stop_name = 'Newcastle Coach Station'),
  8    0
  9  );

1 row created.

w24041293 > INSERT INTO route_stop (route_id, stop_sequence, stop_id, duration_min)
  2  VALUES (
  3    (SELECT route_id FROM route
  4     WHERE route_name = 'Newcastle - Berwick Day Tour'),
  5    2,
  6    (SELECT stop_id FROM stop
  7     WHERE stop_name = 'Morpeth Bus Station'),
  8    20
  9  );

1 row created.
```

```
w24041293 > INSERT INTO route_stop (route_id, stop_sequence, stop_id, duration_min)
  2  VALUES (
  3    (SELECT route_id FROM route
  4     WHERE route_name = 'Newcastle - Berwick Day Tour'),
  5    3,
  6    (SELECT stop_id FROM stop
  7     WHERE stop_name = 'Berwick-upon-Tweed Bus Station'),
  8    0
  9  );

1 row created.

w24041293 > INSERT INTO route_stop (route_id, stop_sequence, stop_id, duration_min)
  2  VALUES (
  3    (SELECT route_id FROM route
  4     WHERE route_name = 'Newcastle - Durham Commuter'),
  5    1,
  6    (SELECT stop_id FROM stop
  7     WHERE stop_name = 'Newcastle Coach Station'),
  8    0
  9  );

1 row created.
```

```
w24041293 > INSERT INTO route_stop (route_id, stop_sequence, stop_id, duration_min)
  2  VALUES (
  3    (SELECT route_id FROM route
  4     WHERE route_name = 'Newcastle - Durham Commuter'),
  5    2,
  6    (SELECT stop_id FROM stop
  7     WHERE stop_name = 'Durham Bus Station'),
  8    0
  9  );

1 row created.
```

```
w24041293 > INSERT INTO route_stop (route_id, stop_sequence, stop_id, duration_min)
  2  VALUES (
  3    (SELECT route_id FROM route
  4     WHERE route_name = 'Edinburgh - Berwick Coastal'),
  5    1,
  6    (SELECT stop_id FROM stop
  7     WHERE stop_name = 'Edinburgh Bus Station'),
  8    0
  9  );

1 row created.

w24041293 > INSERT INTO route_stop (route_id, stop_sequence, stop_id, duration_min)
  2  VALUES (
  3    (SELECT route_id FROM route
  4     WHERE route_name = 'Edinburgh - Berwick Coastal'),
  5    2,
  6    (SELECT stop_id FROM stop
  7     WHERE stop_name = 'Berwick-upon-Tweed Bus Station'),
  8    0
  9  );

1 row created.
```

```
w24041293 > INSERT INTO route_stop (route_id, stop_sequence, stop_id, duration_min)
  2  VALUES (
  3    (SELECT route_id FROM route
  4     WHERE route_name = 'York - Newcastle Express'),
  5    1,
  6    (SELECT stop_id FROM stop
  7     WHERE stop_name = 'York Coach Station'),
  8    0
  9  );

1 row created.

w24041293 > INSERT INTO route_stop (route_id, stop_sequence, stop_id, duration_min)
  2  VALUES (
  3    (SELECT route_id FROM route
  4     WHERE route_name = 'York - Newcastle Express'),
  5    2,
  6    (SELECT stop_id FROM stop
  7     WHERE stop_name = 'Newcastle Coach Station'),
  8    0
  9  );

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO schedule (route_id, coach_id, driver_id,
  2                       departure_datetime, ticket_price,
  3                       seats_available, service_status)
  4  VALUES (
  5    (SELECT route_id FROM route
  6       WHERE route_name = 'Newcastle - Berwick Day Tour'),
  7    (SELECT coach_id FROM coach
  8       WHERE registration_no = 'NT01 NCL'),
  9    (SELECT d.person_id FROM driver d
 10       JOIN person p ON d.person_id = p.person_id
 11      WHERE p.first_name = 'Aditi' AND p.last_name = 'Sharma'),
 12    TO_DATE('2025-12-02 09:00','YYYY-MM-DD HH24:MI'),
 13    25.00,
 14    40,
 15    'scheduled'
 16  );

1 row created.

w24041293 > INSERT INTO schedule (route_id, coach_id, driver_id,
  2                       departure_datetime, ticket_price,
  3                       seats_available, service_status)
  4  VALUES (
  5    (SELECT route_id FROM route
  6       WHERE route_name = 'Newcastle - Berwick Day Tour'),
  7    (SELECT coach_id FROM coach
  8       WHERE registration_no = 'NT01 NCL'),
  9    (SELECT d.person_id FROM driver d
 10       JOIN person p ON d.person_id = p.person_id
 11      WHERE p.first_name = 'Emily' AND p.last_name = 'Watson'),
 12    TO_DATE('2025-12-03 09:00','YYYY-MM-DD HH24:MI'),
 13    25.00,
 14    5,
 15    'scheduled'
 16  );

1 row created.

w24041293 > INSERT INTO schedule (route_id, coach_id, driver_id,
  2                       departure_datetime, ticket_price,
  3                       seats_available, service_status)
  4  VALUES (
  5    (SELECT route_id FROM route
  6       WHERE route_name = 'Newcastle - Berwick Day Tour'),
  7    (SELECT coach_id FROM coach
  8       WHERE registration_no = 'NT02 DUR'),
  9    (SELECT d.person_id FROM driver d
 10       JOIN person p ON d.person_id = p.person_id
 11      WHERE p.first_name = 'Michael' AND p.last_name = 'Carter'),
 12    TO_DATE('2025-12-05 14:00','YYYY-MM-DD HH24:MI'),
 13    27.00,
 14    18,
 15    'scheduled'
 16  );

1 row created.
```

```
w24041293 > INSERT INTO schedule (route_id, coach_id, driver_id,
  2                       departure_datetime, ticket_price,
  3                       seats_available, service_status)
  4  VALUES (
  5    (SELECT route_id FROM route
  6       WHERE route_name = 'Newcastle - Durham Commuter'),
  7    (SELECT coach_id FROM coach
  8       WHERE registration_no = 'NT02 DUR'),
  9    (SELECT d.person_id FROM driver d
 10       JOIN person p ON d.person_id = p.person_id
 11      WHERE p.first_name = 'Emily' AND p.last_name = 'Watson'),
 12    TO_DATE('2025-12-02 08:00','YYYY-MM-DD HH24:MI'),
 13    15.00,
 14    30,
 15    'scheduled'
 16  );

1 row created.

w24041293 > INSERT INTO schedule (route_id, coach_id, driver_id,
  2                       departure_datetime, ticket_price,
  3                       seats_available, service_status)
  4  VALUES (
  5    (SELECT route_id FROM route
  6       WHERE route_name = 'Edinburgh - Berwick Coastal'),
  7    (SELECT coach_id FROM coach
  8       WHERE registration_no = 'NT03 EDI'),
  9    (SELECT d.person_id FROM driver d
 10       JOIN person p ON d.person_id = p.person_id
 11      WHERE p.first_name = 'Michael' AND p.last_name = 'Carter'),
 12    TO_DATE('2025-12-04 10:00','YYYY-MM-DD HH24:MI'),
 13    30.00,
 14    10,
 15    'scheduled'
 16  );

1 row created.

w24041293 > INSERT INTO schedule (route_id, coach_id, driver_id,
  2                       departure_datetime, ticket_price,
  3                       seats_available, service_status)
  4  VALUES (
  5    (SELECT route_id FROM route
  6       WHERE route_name = 'York - Newcastle Express'),
  7    (SELECT coach_id FROM coach
  8       WHERE registration_no = 'NT04 YORK'),
  9    (SELECT d.person_id FROM driver d
 10       JOIN person p ON d.person_id = p.person_id
 11      WHERE p.first_name = 'Liam' AND p.last_name = 'Brown'),
 12    TO_DATE('2025-12-10 07:30','YYYY-MM-DD HH24:MI'),
 13    20.00,
 14    25,
 15    'scheduled'
 16  );

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO reservation (agent_id, customer_id, reservation_date,
  2                          requested_seats, reservation_status)
  3  VALUES (
  4    (SELECT agent_id FROM agent
  5     WHERE name = 'Northern Travels Newcastle'),
  6    (SELECT customer_id FROM customer
  7     WHERE name = 'Priya Singh'),
  8    TO_DATE('2025-09-15','YYYY-MM-DD'),
  9    2,
 10    'confirmed'
 11  );

1 row created.

w24041293 > INSERT INTO reservation (agent_id, customer_id, reservation_date,
  2                          requested_seats, reservation_status)
  3  VALUES (
  4    (SELECT agent_id FROM agent
  5     WHERE name = 'Northern Travels Newcastle'),
  6    (SELECT customer_id FROM customer
  7     WHERE name = 'Oliver Green'),
  8    TO_DATE('2025-09-16','YYYY-MM-DD'),
  9    1,
 10    'confirmed'
 11  );

1 row created.

w24041293 > INSERT INTO reservation (agent_id, customer_id, reservation_date,
  2                          requested_seats, reservation_status)
  3  VALUES (
  4    (SELECT agent_id FROM agent
  5     WHERE name = 'Northern Travels Newcastle'),
  6    (SELECT customer_id FROM customer
  7     WHERE name = 'Sophia Miller'),
  8    TO_DATE('2025-10-01','YYYY-MM-DD'),
  9    1,
 10    'confirmed'
 11  );

1 row created.

w24041293 > INSERT INTO reservation (agent_id, customer_id, reservation_date,
  2                          requested_seats, reservation_status)
  3  VALUES (
  4    (SELECT agent_id FROM agent
  5     WHERE name = 'Durham City Breaks'),
  6    (SELECT customer_id FROM customer
  7     WHERE name = 'Arjun Rao'),
  8    TO_DATE('2025-09-20','YYYY-MM-DD'),
  9    2,
 10    'confirmed'
 11  );

1 row created.
```

```
w24041293 > INSERT INTO reservation (agent_id, customer_id, reservation_date,
  2                          requested_seats, reservation_status)
  3  VALUES (
  4    (SELECT agent_id FROM agent
  5     WHERE name = 'Global Trips London'),
  6    (SELECT customer_id FROM customer
  7     WHERE name = 'Chloe Anderson'),
  8    TO_DATE('2025-10-05','YYYY-MM-DD'),
  9    1,
 10    'confirmed'
 11  );

1 row created.

w24041293 > INSERT INTO reservation (agent_id, customer_id, reservation_date,
  2                          requested_seats, reservation_status)
  3  VALUES (
  4    (SELECT agent_id FROM agent
  5     WHERE name = 'Sunshine Tours Mumbai'),
  6    (SELECT customer_id FROM customer
  7     WHERE name = 'Mia Patel'),
  8    TO_DATE('2025-09-01','YYYY-MM-DD'),
  9    1,
 10    'cancelled'
 11  );

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO ticket (reservation_id, schedule_id, issue_date,
  2                       seat_allocation, price, status)
  3  VALUES (
  4    (SELECT reservation_id FROM reservation
  5      WHERE reservation_date = TO_DATE('2025-09-15','YYYY-MM-DD')
  6        AND customer_id = (SELECT customer_id FROM customer
  7                           WHERE name = 'Priya Singh')),
  8    (SELECT schedule_id FROM schedule s
  9       JOIN route r ON s.route_id = r.route_id
 10      WHERE r.route_name = 'Newcastle - Berwick Day Tour'
 11        AND s.departure_datetime = TO_DATE('2025-12-02 09:00',
 12                                           'YYYY-MM-DD HH24:MI')),
 13    TO_DATE('2025-09-15','YYYY-MM-DD'),
 14    '1A',
 15    25.00,
 16    'confirmed'
 17  );

1 row created.

w24041293 > INSERT INTO ticket (reservation_id, schedule_id, issue_date,
  2                       seat_allocation, price, status)
  3  VALUES (
  4    (SELECT reservation_id FROM reservation
  5      WHERE reservation_date = TO_DATE('2025-09-15','YYYY-MM-DD')
  6        AND customer_id = (SELECT customer_id FROM customer
  7                           WHERE name = 'Priya Singh')),
  8    (SELECT schedule_id FROM schedule s
  9       JOIN route r ON s.route_id = r.route_id
 10      WHERE r.route_name = 'Newcastle - Berwick Day Tour'
 11        AND s.departure_datetime = TO_DATE('2025-12-02 09:00',
 12                                           'YYYY-MM-DD HH24:MI')),
 13    TO_DATE('2025-09-15','YYYY-MM-DD'),
 14    '1B',
 15    25.00,
 16    'confirmed'
 17  );

1 row created.

w24041293 > INSERT INTO ticket (reservation_id, schedule_id, issue_date,
  2                       seat_allocation, price, status)
  3  VALUES (
  4    (SELECT reservation_id FROM reservation
  5      WHERE reservation_date = TO_DATE('2025-09-16','YYYY-MM-DD')
  6        AND customer_id = (SELECT customer_id FROM customer
  7                           WHERE name = 'Oliver Green')),
  8    (SELECT schedule_id FROM schedule s
  9       JOIN route r ON s.route_id = r.route_id
 10      WHERE r.route_name = 'Newcastle - Berwick Day Tour'
 11        AND s.departure_datetime = TO_DATE('2025-12-05 14:00',
 12                                           'YYYY-MM-DD HH24:MI')),
 13    TO_DATE('2025-09-16','YYYY-MM-DD'),
 14    '3C',
 15    27.00,
 16    'confirmed'
 17  );

1 row created.
```

```
w24041293 > INSERT INTO ticket (reservation_id, schedule_id, issue_date,
  2                       seat_allocation, price, status)
  3  VALUES (
  4    (SELECT reservation_id FROM reservation
  5      WHERE reservation_date = TO_DATE('2025-10-01','YYYY-MM-DD')
  6        AND customer_id = (SELECT customer_id FROM customer
  7                           WHERE name = 'Sophia Miller')),
  8    (SELECT schedule_id FROM schedule s
  9       JOIN route r ON s.route_id = r.route_id
 10      WHERE r.route_name = 'York - Newcastle Express'
 11        AND s.departure_datetime = TO_DATE('2025-12-10 07:30',
 12                                           'YYYY-MM-DD HH24:MI')),
 13    TO_DATE('2025-10-01','YYYY-MM-DD'),
 14    '5D',
 15    20.00,
 16    'confirmed'
 17  );

1 row created.
```

```
w24041293 > INSERT INTO ticket (reservation_id, schedule_id, issue_date,
  2                       seat_allocation, price, status)
  3  VALUES (
  4    (SELECT reservation_id FROM reservation
  5      WHERE reservation_date = TO_DATE('2025-09-20','YYYY-MM-DD')
  6        AND customer_id = (SELECT customer_id FROM customer
  7                           WHERE name = 'Arjun Rao')),
  8    (SELECT schedule_id FROM schedule s
  9       JOIN route r ON s.route_id = r.route_id
 10      WHERE r.route_name = 'Newcastle - Durham Commuter'
 11        AND s.departure_datetime = TO_DATE('2025-12-02 08:00',
 12                                           'YYYY-MM-DD HH24:MI')),
 13    TO_DATE('2025-09-20','YYYY-MM-DD'),
 14    '2A',
 15    15.00,
 16    'confirmed'
 17  );

1 row created.
```

```
w24041293 > INSERT INTO ticket (reservation_id, schedule_id, issue_date,
  2                       seat_allocation, price, status)
  3  VALUES (
  4    (SELECT reservation_id FROM reservation
  5      WHERE reservation_date = TO_DATE('2025-09-20','YYYY-MM-DD')
  6        AND customer_id = (SELECT customer_id FROM customer
  7                           WHERE name = 'Arjun Rao')),
  8    (SELECT schedule_id FROM schedule s
  9       JOIN route r ON s.route_id = r.route_id
 10      WHERE r.route_name = 'Newcastle - Durham Commuter'
 11        AND s.departure_datetime = TO_DATE('2025-12-02 08:00',
 12                                           'YYYY-MM-DD HH24:MI')),
 13    TO_DATE('2025-09-20','YYYY-MM-DD'),
 14    '2B',
 15    15.00,
 16    'confirmed'
 17  );

1 row created.
```

```
w24041293 > INSERT INTO ticket (reservation_id, schedule_id, issue_date,
  2                     seat_allocation, price, status)
  3  VALUES (
  4    (SELECT reservation_id FROM reservation
  5      WHERE reservation_date = TO_DATE('2025-10-05','YYYY-MM-DD')
  6        AND customer_id = (SELECT customer_id FROM customer
  7                           WHERE name = 'Chloe Anderson')),
  8    (SELECT schedule_id FROM schedule s
  9       JOIN route r ON s.route_id = r.route_id
 10      WHERE r.route_name = 'Edinburgh - Berwick Coastal'
 11        AND s.departure_datetime = TO_DATE('2025-12-04 10:00',
 12                                           'YYYY-MM-DD HH24:MI')),
 13    TO_DATE('2025-10-05','YYYY-MM-DD'),
 14    '4A',
 15    30.00,
 16    'confirmed'
 17  );

1 row created.

w24041293 > INSERT INTO ticket (reservation_id, schedule_id, issue_date,
  2                     seat_allocation, price, status)
  3  VALUES (
  4    (SELECT reservation_id FROM reservation
  5      WHERE reservation_date = TO_DATE('2025-09-01','YYYY-MM-DD')
  6        AND customer_id = (SELECT customer_id FROM customer
  7                           WHERE name = 'Mia Patel')),
  8    (SELECT schedule_id FROM schedule s
  9       JOIN route r ON s.route_id = r.route_id
 10      WHERE r.route_name = 'Newcastle - Berwick Day Tour'
 11        AND s.departure_datetime = TO_DATE('2025-12-03 09:00',
 12                                           'YYYY-MM-DD HH24:MI')),
 13    TO_DATE('2025-09-01','YYYY-MM-DD'),
 14    '6C',
 15    25.00,
 16    'cancelled'
 17  );

1 row created.

w24041293 > SPOOL OFF;
```

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

(8 marks)

q1) 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):

q2) 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;
```

```
w24041293 > SPOOL OFF;
w24041293 > SELECT s.schedule_id,
  2         r.route_name,
  3         s.departure_datetime,
  4         s.ticket_price,
  5         s.seats_available,
  6         s.service_status
  7  FROM   schedule s
  8         JOIN route r
  9           ON s.route_id = r.route_id
 10  WHERE  r.route_name = 'Newcastle - Berwick Day Tour'
 11    AND  s.seats_available >= 7
 12    AND  s.departure_datetime BETWEEN TRUNC(SYSDATE)
 13                                  AND TRUNC(SYSDATE) + 14
 14  ORDER BY s.departure_datetime;

SCHEDULE_ID
-----------
ROUTE_NAME
--------------------------------------------------------------------------------
DEPARTURE TICKET_PRICE SEATS_AVAILABLE SERVICE_STATUS
--------- ------------ --------------- --------------------
         25
Newcastle - Berwick Day Tour
02-DEC-25           25              40 scheduled

         27
Newcastle - Berwick Day Tour
05-DEC-25           27              18 scheduled

SCHEDULE_ID
-----------
ROUTE_NAME
--------------------------------------------------------------------------------
DEPARTURE TICKET_PRICE SEATS_AVAILABLE SERVICE_STATUS
--------- ------------ --------------- --------------------


w24041293 >
```

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;
```

```
w24041293 > SELECT agent_id,
  2         agent_name,
  3         tickets_sold
  4  FROM   (
  5           SELECT a.agent_id,
  6                  a.name AS agent_name,
  7                  COUNT(t.ticket_id) AS tickets_sold
  8           FROM   agent a
  9                  JOIN reservation r
 10                    ON r.agent_id = a.agent_id
 11                  JOIN ticket t
 12                    ON t.reservation_id = r.reservation_id
 13           WHERE  t.issue_date BETWEEN DATE '2025-09-15' AND DATE '2025-10-15'
 14             AND  t.status = 'confirmed'
 15           GROUP BY a.agent_id, a.name
 16         ) ticket_per_agent
 17  WHERE  tickets_sold = (
 18           SELECT MAX(tickets_sold)
 19           FROM   (
 20                    SELECT COUNT(t2.ticket_id) AS tickets_sold
 21                    FROM   agent a2
 22                           JOIN reservation r2
 23                             ON r2.agent_id = a2.agent_id
 24                           JOIN ticket t2
 25                             ON t2.reservation_id = r2.reservation_id
 26                    WHERE  t2.issue_date BETWEEN DATE '2025-09-15' AND DATE '2025-10-15'
 27                      AND  t2.status = 'confirmed'
 28                    GROUP BY a2.agent_id, a2.name
 29                  )
 30         )
 31  ORDER BY agent_name;

  AGENT_ID
----------
AGENT_NAME
--------------------------------------------------------------------------------
TICKETS_SOLD
------------
        23
Northern Travels Newcastle
           4
```

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)

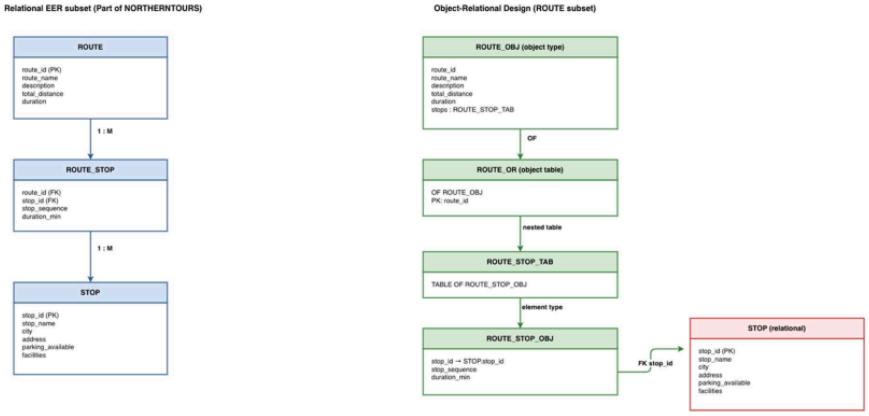
1
For the object-relational part I chose the route planning subset of my NORTHERNTOURS conceptual design. This subset includes the entities:

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.

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



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.

3) 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;
```

```
w24041293 > @D:\northerntours\scripts\northerntours_create_tables_objectrelational.sql
w24041293 > CREATE TYPE ROUTE_STOP_OBJ AS OBJECT (
  2      stop_id        NUMBER(10),
  3      stop_sequence  NUMBER(3),
  4      duration_min   NUMBER(5)
  5  );
  6  /

Type created.

w24041293 >
w24041293 > CREATE TYPE ROUTE_STOP_TAB AS TABLE OF ROUTE_STOP_OBJ;
  2  /

Type created.

w24041293 >
w24041293 > CREATE TYPE ROUTE_OBJ AS OBJECT (
  2      route_id       NUMBER(10),
  3      route_name     VARCHAR2(100),
  4      description    VARCHAR2(200),
  5      total_distance NUMBER(6,1),
  6      duration       NUMBER(5),
  7      stops          ROUTE_STOP_TAB
  8  );
  9  /

Type created.

w24041293 >
w24041293 > CREATE TABLE ROUTE_OR OF ROUTE_OBJ (
  2      CONSTRAINT pk_route_or PRIMARY KEY (route_id)
  3  )
  4  NESTED TABLE stops
  5      STORE AS route_or_stops;

Table created.
```

4) 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
   )
  )
 )
);
```

```
w24041293 > @D:\northerntours\scripts\northerntours_create_tables_output_insertObjRel.sql
w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities)
  2  VALUES ('Newcastle Coach Station',
  3          'Newcastle upon Tyne',
  4          'John Dobson St, Newcastle',
  5          'Y',
  6          'toilets; caf├®; Wi-Fi');

1 row created.

w24041293 >
w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities)
  2  VALUES ('Morpeth Bus Station',
  3          'Morpeth',
  4          'Bridge St, Morpeth',
  5          'Y',
  6          'toilets; kiosk');

1 row created.

w24041293 >
w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities)
  2  VALUES ('Alnwick Castle Stop',
  3          'Alnwick',
  4          'Castle Square, Alnwick',
  5          'N',
  6          'photo spot; ticket office nearby');

1 row created.

w24041293 >
w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities)
  2  VALUES ('Berwick-upon-Tweed Coach Park',
  3          'Berwick-upon-Tweed',
  4          'Ramparts Walk, Berwick',
  5          'Y',
  6          'toilets; caf├®; viewpoint');

1 row created.

w24041293 >
w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities)
  2  VALUES ('Durham Coach Park',
  3          'Durham',
  4          'Freeman''s Place, Durham',
  5          'Y',
  6          'toilets; caf├®; city centre access');

1 row created.

w24041293 >
w24041293 > INSERT INTO stop (stop_name, city, address, parking_available, facilities)
  2  VALUES ('York Rail Station',
  3          'York',
  4          'Station Rd, York',
  5          'N',
  6          'toilets; shops; connections');

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO route_or VALUES (
  2    ROUTE_OBJ(
  3      9001,
  4      'Newcastle – Berwick Coastal Explorer',
  5      'Day tour along the Northumberland coast with photo stops.',
  6      120.0,
  7      420,
  8      ROUTE_STOP_TAB(
  9        ROUTE_STOP_OBJ(
 10          (SELECT stop_id FROM stop WHERE stop_name = 'Newcastle Coach Station'),
 11          1,
 12          60
 13        ),
 14        ROUTE_STOP_OBJ(
 15          (SELECT stop_id FROM stop WHERE stop_name = 'Morpeth Bus Station'),
 16          2,
 17          60
 18        ),
 19        ROUTE_STOP_OBJ(
 20          (SELECT stop_id FROM stop WHERE stop_name = 'Alnwick Castle Stop'),
 21          3,
 22          90
 23        ),
 24        ROUTE_STOP_OBJ(
 25          (SELECT stop_id FROM stop WHERE stop_name = 'Berwick-upon-Tweed Coach Park'),
 26          4,
 27          210
 28        )
 29      )
 30    )
 31  );

1 row created.

w24041293 >
w24041293 > INSERT INTO route_or VALUES (
  2    ROUTE_OBJ(
  3      9002,
  4      'Newcastle – Durham City Highlights',
  5      'Short return trip from Newcastle to Durham with free time in the city.',
  6      50.0,
  7      240,
  8      ROUTE_STOP_TAB(
  9        ROUTE_STOP_OBJ(
 10          (SELECT stop_id FROM stop WHERE stop_name = 'Newcastle Coach Station'),
 11          1,
 12          75
 13        ),
 14        ROUTE_STOP_OBJ(
 15          (SELECT stop_id FROM stop WHERE stop_name = 'Durham Coach Park'),
 16          2,
 17          165
 18        )
 19      )
 20    )
 21  );

1 row created.
```

```
w24041293 >
w24041293 > INSERT INTO route_or VALUES (
  2    ROUTE_OBJ(
  3      9003,
  4      'Newcastle – York via Durham',
  5      'Full-day tour visiting Durham and York in one trip.',
  6      160.0,
  7      540,
  8      ROUTE_STOP_TAB(
  9        ROUTE_STOP_OBJ(
 10          (SELECT stop_id FROM stop WHERE stop_name = 'Newcastle Coach Station'),
 11          1,
 12          60
 13        ),
 14        ROUTE_STOP_OBJ(
 15          (SELECT stop_id FROM stop WHERE stop_name = 'Durham Coach Park'),
 16          2,
 17          120
 18        ),
 19        ROUTE_STOP_OBJ(
 20          (SELECT stop_id FROM stop WHERE stop_name = 'York Rail Station'),
 21          3,
 22          360
 23        )
 24      )
 25    )
 26  );

1 row created.

w24041293 >
w24041293 > INSERT INTO route_or VALUES (
  2    ROUTE_OBJ(
  3      9004,
  4      'Durham – Alnwick Castle Day Trip',
  5      'Coach from Durham to Alnwick Castle and back.',
  6      110.0,
  7      420,
  8      ROUTE_STOP_TAB(
  9        ROUTE_STOP_OBJ(
 10          (SELECT stop_id FROM stop WHERE stop_name = 'Durham Coach Park'),
 11          1,
 12          90
 13        ),
 14        ROUTE_STOP_OBJ(
 15          (SELECT stop_id FROM stop WHERE stop_name = 'Alnwick Castle Stop'),
 16          2,
 17          240
 18        ),
 19        ROUTE_STOP_OBJ(
 20          (SELECT stop_id FROM stop WHERE stop_name = 'Durham Coach Park'),
 21          3,
 22          90
 23        )
 24      )
 25    )
 26  );

1 row created.
```

5) 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;
```

```
w24041293 > SELECT
  2      r.route_id,
  3      r.route_name,
  4      COUNT(s.stop_id)    AS number_of_stops,
  5      SUM(s.duration_min) AS total_stop_duration_min
  6  FROM
  7      route_or r,
  8      TABLE(r.stops) s
  9  GROUP BY
 10      r.route_id,
 11      r.route_name
 12  ORDER BY
 13      r.route_id;

  ROUTE_ID
----------
ROUTE_NAME
--------------------------------------------------------------------------------
NUMBER_OF_STOPS TOTAL_STOP_DURATION_MIN
--------------- -----------------------
      9001
Newcastle - Berwick Coastal Explorer
              4                     420

      9002
Newcastle - Durham City Highlights
              2                     240

  ROUTE_ID
----------
ROUTE_NAME
--------------------------------------------------------------------------------
NUMBER_OF_STOPS TOTAL_STOP_DURATION_MIN
--------------- -----------------------

      9003
Newcastle - York via Durham
              3                     540

      9004
Durham - Alnwick Castle Day Trip

  ROUTE_ID
----------
ROUTE_NAME
--------------------------------------------------------------------------------
NUMBER_OF_STOPS TOTAL_STOP_DURATION_MIN
--------------- -----------------------
              3                     420


w24041293 >
```

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;
```

```
w24041293 > SELECT
  2      r.route_id,
  3      r.route_name,
  4      st.city,
  5      COUNT(*) AS times_in_city
  6  FROM
  7      route_or r,
  8      TABLE(r.stops) s,
  9      stop st
 10  WHERE
 11      s.stop_id = st.stop_id
 12      AND st.city = 'Durham'
 13  GROUP BY
 14      r.route_id,
 15      r.route_name,
 16      st.city
 17  ORDER BY
 18      times_in_city DESC,
 19      r.route_name;

  ROUTE_ID
----------
ROUTE_NAME
--------------------------------------------------------------------------------
CITY
--------------------------------------------------------------------------------
TIMES_IN_CITY
-------------
      9004
Durham - Alnwick Castle Day Trip
Durham
            2


  ROUTE_ID
----------
ROUTE_NAME
--------------------------------------------------------------------------------
CITY
--------------------------------------------------------------------------------
TIMES_IN_CITY
-------------
      9002
Newcastle - Durham City Highlights
Durham
            1


  ROUTE_ID
----------
ROUTE_NAME
--------------------------------------------------------------------------------
CITY
--------------------------------------------------------------------------------
TIMES_IN_CITY
-------------
      9003
Newcastle - York via Durham
Durham
            1
```

(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.

2) 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.

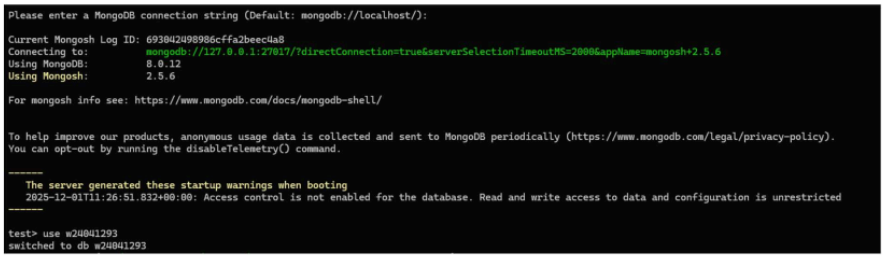
This design is denormalised with respect to the relational schema, but it reflects how the business actually looks at the data: one full sales list per agent per day. It is easy to answer typical questions in a straightforward way, such as “how many tickets did each agent sell in one month?” or “what 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.

3) 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'
      }
    }
  ]
 }
]);
```



```
w24041293> load("D:/northerntours/scripts/northerntours_nosql_agent_sales.js")
true
w24041293> db.agent_sales.find().pretty()
[
  {
    _id: ObjectId('693043798986cffa2beec4ac'),
    agent_id: 201,
    agent_name: 'Tyne Travel Newcastle',
    agent_city: 'Newcastle upon Tyne',
    sales_date: ISODate('2025-09-15T00:00:00.000Z'),
    sales: [
      {
        sale_id: ObjectId('693043798986cffa2beec4a9'),
        schedule_id: 301,
        route_id: 9001,
        route_description: 'Newcastle – Berwick Coastal Explorer',
        departure_datetime: ISODate('2025-09-18T08:30:00.000Z'),
        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('693043798986cffa2beec4aa'),
        schedule_id: 302,
        route_id: 9002,
        route_description: 'Newcastle – Durham City Highlights',
        departure_datetime: ISODate('2025-09-19T09:00:00.000Z'),
        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'
        }
      }
    ]
  },
```

```
  {
    _id: ObjectId('693043798986cffa2beec4ad'),
    agent_id: 202,
    agent_name: 'Wearside Holiday Agents',
    agent_city: 'Sunderland',
    sales_date: ISODate('2025-09-15T00:00:00.000Z'),
    sales: [
      {
        sale_id: ObjectId('693043798986cffa2beec4ab'),
        schedule_id: 305,
        route_id: 9003,
        route_description: 'Newcastle – York via Durham',
        departure_datetime: ISODate('2025-09-20T07:45:00.000Z'),
        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'
        }
      }
    ]
  }
]
```

```
// 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 } }
]);
```

```
w24041293> 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 } }
... ]);
...
[
  {
    _id: { agent_id: 201, agent_name: 'Tyne Travel Newcastle' },
    tickets_sold: 3,
    total_revenue: 120
  },
  {
    _id: { agent_id: 202, agent_name: 'Wearside Holiday Agents' },
    tickets_sold: 1,
    total_revenue: 60
  }
]
w24041293>
```

```
// 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
 }
);
```

```
w24041293> db.agent_sales.find(
...   {
...     'sales.route_id': 9003,
...     'sales.tickets.status': 'cancelled'
...   },
...   {
...     agent_name: 1,
...     agent_city: 1,
...     sales_date: 1,
...     'sales.$': 1
...   }
... );
...
[
  {
    _id: ObjectId('693043798986cffa2beec4ad'),
    agent_name: 'Wearside Holiday Agents',
    agent_city: 'Sunderland',
    sales_date: ISODate('2025-09-15T00:00:00.000Z'),
    sales: [
      {
        sale_id: ObjectId('693043798986cffa2beec4ab'),
        schedule_id: 305,
        route_id: 9003,
        route_description: 'Newcastle – York via Durham',
        departure_datetime: ISODate('2025-09-20T07:45:00.000Z'),
        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'
        }
      }
    ]
  }
]
w24041293>
```

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

KL7011_w24041293.docx

ORIGINALITY REPORT

27%	11%	0%	27%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Northumbria at Newcastle Student Paper	18%
2	syl.gautier.online.fr Internet Source	3%
3	Submitted to University of London Worldwide Student Paper	2%
4	Submitted to University College London Student Paper	1%
5	Submitted to University of South Australia Student Paper	1%
6	Submitted to Murdoch University Student Paper	1%
7	Submitted to De Montfort University Student Paper	1%
8	Submitted to University of Northampton Student Paper	<1%
9	Submitted to Federation University Student Paper	<1%

10	www.coursehero.com Internet Source	<1%
11	cherryberrykix66.github.io Internet Source	<1%
12	Submitted to Asia Pacific Instutute of Information Technology Student Paper	<1%
13	itcareershift.com Internet Source	<1%
14	ebin.pub Internet Source	<1%
15	Submitted to Leeds Beckett University Student Paper	<1%

Exclude quotes	On	Exclude assignment template	On
Exclude bibliography	On	Exclude matches	Off