

Session 4

Data Flow Diagrams, Entity Relationship Diagram, and Data Dictionary for the Railway Reservation System

This session turns the SRS of Session 3 into three analysis models of the Railway Reservation System. The Data Flow Diagrams (DFDs) show how data moves between the actors, the eight modules, and the data stores. The Entity Relationship Diagram (ERD) shows what data the system keeps and how the entities relate. The Data Dictionary defines every attribute and every data flow so that two people reading the diagrams mean the same thing. Together they are the functional model and the data model that the manual asks for in the SRS appendices, and they are the input for the modular design in Session 5.

Objectives

✕ Do not copy. Read for understanding and the viva

- Draw a context diagram (level 0) with the RRS as one process and five external entities.
- Decompose it into a level 1 DFD whose processes 1.0 to 8.0 are the eight modules.
- Decompose Booking (4.0) into a level 2 DFD, balanced with level 1.
- Draw the ERD of the twelve RRS entities with cardinality and participation.
- Write a data dictionary for every entity and for the main data flows.

Problem Statement

■ Write in lab record

- (a) Draw DFDs up to appropriate levels for the RRS.
- (b) Draw ERDs for the RRS. Describe the relationships between different entities.
- (c) Design Data Dictionary for RRS.

Concept

✗ Do not copy. Read for understanding and the viva

DFD symbols and the text notation

A DFD has four symbols: external entity (a rectangle, a source or sink of data outside the system), process (a circle or rounded box, numbered, named with a verb), data store (an open rectangle, named with a noun, labelled D1, D2, and so on), and data flow (an arrow with a name). On this page the text form is: external entities in plain boxes, processes as `[n.0 Verb Object]`, stores as `Dn | Name`, flows as labelled arrows. Redraw them by hand with the standard symbols.

Levelling and balancing

Level 0 (the context diagram) shows one process, the whole system, and every external entity. Level 1 explodes that one process into the main processes; here one per module, so 1.0 to 8.0. Level 2 explodes one level 1 process into sub-processes 4.1, 4.2, and so on. The rule of balancing: the flows entering and leaving a process at level n must be exactly the flows entering and leaving its explosion at level $n+1$. If level 1 shows "booking request" going into 4.0, then some 4.x sub-process must receive "booking request".

DFD rules that examiners check

- A data flow never joins two external entities, two data stores, or an entity and a store directly. A process must be in between.
- Every process has at least one input and one output.
- Names on flows are data, not actions: "booking request", not "book".
- Stores appear at level 1 and below, never on the context diagram.

ERD notation

An entity is a rectangle; a relationship is a diamond or a labelled line; an attribute is listed under the entity with the primary key underlined (shown here as `PK`). Cardinality says how many instances of one entity relate to one instance of the other: 1:1, 1:M, or M:N. Participation says whether every instance must take part (total, double line) or may (partial, single line). A M:N relationship is resolved with an associative entity; in the RRS, Route resolves Train to Station.

Data dictionary notation

The dictionary defines attributes (name, type, size, constraints, meaning) and data flows (composition). Flow composition uses `=` for "is composed of", `+` for "and", `(x)` for optional, and `n..m of x` for repetition. Every name in a DFD or ERD must appear in the dictionary once, with one meaning.

Data Flow Diagrams

■ Write in lab record

Context diagram (level 0)

```

1          registration, login, search criteria,
2          booking request, cancellation request, PNR enquiry
3  +-----+ +-----+-----> +-----+
4  | Passenger |                                     |
5  +-----+ <-----+-----+ |
6          train list with availability and fare,   |
7          ticket (PNR), PNR status, refund details |
8                                     |
9  +-----+ counter booking request, cancellation by PNR → |
10 | Reservation |                                     | Railw
11 |   Clerk   | ← ticket (PNR), refund slip ----- |
12 +-----+                                     |
13                                     |
14 +-----+ station, train, route, coach, fare and -----> |
15 | Administrator | schedule details, report request       |
16 +-----+ ← occupancy, revenue, cancellation reports -- |
17                                     +-----+
18                                     | ^
19          payment request,             | |
20          refund instruction           | |
21                                     v |
22                                     +-----+
23                                     | Payment Gatew
24                                     +-----+
25          payment resul

```

Element	Kind	Description
0 Railway Reservation System	Process	The whole system, exploded at level 1
Passenger	External entity	Books online; sends registration, login, search criteria, booking request, cancellation request, PNR enquiry; receives train list, ticket, PNR status, refund details
Reservation Clerk	External entity	Books and cancels at the counter; sends counter booking request and cancellation by PNR; receives ticket and refund slip
Administrator	External entity	Sends station, train, route, coach, fare, schedule details and report requests; receives reports
Payment Gateway	External entity	Receives payment request and refund instruction; returns payment result
Notification Service	External entity	Receives notification request; returns delivery status

Level 1 DFD

```

1  Passenger, Clerk, Admin --registration, login→ [1.0 Manage Users] ←read
2      ←login result-----
3      (1.0 supplies "user role" to 4.0, 5.0 and 8.0; thos
4
5  Administrator --station, train, route, coach,→ [2.0 Manage Trains  --writ
6      fare, schedule details                and Schedules]    --write
7                                          |
8                                          +--schedule change—
9
10 Passenger --search criteria-----> [3.0 Search Trains] ←read
11      ←train list, availability, fare----                ←read
12                                                        ←read
13
14 Passenger, Clerk --booking request, passenger→ [4.0 Book Ticket] ←read,
15      details, PNR enquiry                |      ^      ←rea
16      ←ticket (PNR), PNR status-----    |      |      ←rea
17                                          |      |      (sends
18      payment request|      |payment resul
19                                          v      |
20 Payment Gateway ←payment request,----- [6.0 Process Payment] --wr
21      refund instruction                ^                --
22      --payment result----->          |
23                                          |refund request
24                                          |
25 Passenger, Clerk --cancellation by PNR-----> [5.0 Cancel and Refund] ←
26      ←refund details, refund slip--      |                --
27                                          |
28                                          +--booking event→
29
30 Notification Service ←notification request-- [7.0 Send Notifications] --w
31      --delivery status----->      (inputs: booking event fr
32      schedule change from 2.0
33
34 Administrator --report request-----> [8.0 Generate Reports] ←r
35      ←occupancy, revenue,                ←r
36      cancellation reports                ←r
37                                          ←r

```

Processes:

No	Process	Inputs	Outputs
1.0	Manage Users	registration, login (from Passenger, Clerk, Admin); D1	login result; user role to 4.0, 5.0, 8.0; D1
2.0	Manage Trains and Schedules	station, train, route, coach, fare, schedule details (from Administrator)	D2, D3; schedule change to 7.0
3.0	Search Trains	search criteria (from Passenger); D2, D3, D4	train list with availability and fare
4.0	Book Ticket	booking request, passenger details, PNR enquiry; payment result from 6.0; D2, D3, D4	ticket (PNR), PNR status; payment request to 6.0; booking event to 7.0; D4
5.0	Cancel and Refund	cancellation by PNR; D4	refund details, refund slip; refund request to 6.0; booking event to 7.0; D4, D6
6.0	Process Payment	payment request from 4.0; refund request from 5.0; payment result from Payment Gateway	payment request and refund instruction to Payment Gateway; payment result to 4.0; D5, D6
7.0	Send Notifications	booking event from 4.0 and 5.0; schedule change from 2.0; delivery status from Notification Service	notification request to Notification Service; D7
8.0	Generate Reports	report request (from Administrator); D3, D4, D5, D6	occupancy, revenue, cancellation reports

Data stores:

Store	Contents	Written by	Read by
D1 User	User records with role	1.0	1.0, 4.0, 5.0, 8.0
D2 Train Master	Station, Train, Route, Coach, Seat, Fare	2.0	3.0, 4.0
D3 Schedule	Schedule records with status	2.0	3.0, 4.0, 5.0, 8.0
D4 Booking	Booking and Passenger records, seat allocation, waitlist numbers	4.0, 5.0	3.0, 4.0, 5.0, 8.0
D5 Payment	Payment records with gateway txn_ref	6.0	8.0
D6 Refund	Refund records with processing status	5.0, 6.0	8.0
D7 Notification Log	Sent messages with delivery status	7.0	7.0 (retries)

Data flows between processes:

Flow	From	To	Content
user role	1.0	4.0, 5.0, 8.0	user_id, role for access checks
schedule change	2.0	7.0	schedule_id, new status
payment request	4.0	6.0	pnr (provisional), amount, mode
payment result	6.0	4.0	payment_id, status, txn_ref
refund request	5.0	6.0	refund_id, pnr, amount, original mode
booking event	4.0, 5.0	7.0	pnr, event type, user contact, details

Level 2 DFD: 4.0 Book Ticket

```

1  Passenger, Clerk ---booking request (schedule_id,---> [4.1 Validate Request
2      class, from_station, to_station) |
3      ←rejection (departed, not scheduled, |
4      not logged in)----- | valid request
5                               v
6  Passenger, Clerk ---passenger details (1 to 6)-----> [4.2 Capture Passenger
7      ←rejection (more than 6)----- |
8                               | passenger list
9                               v
10                             [4.3 Compute Fare] <-
11                             |
12                             | fare (per passenger)
13                             v
14                             [4.4 Allocate Seats <
15                             or Waitlist] <-
16                             |
17                             | allocation (seats)
18                             v
19                             [4.5 Generate PNR and
20                             Record Booking]
21                             | ^ |
22                             payment request | | payment |
23                             v | result v
24                             6.0 Process ticket (P
25                             Payment to Passenger
26                             Clerk
27
28  Passenger ---PNR enquiry---> [4.5 Generate PNR and Record Booking] ---PNR status
29      (reads D4 | Booking)

```

No	Process	Inputs	Outputs
4.1	Validate Request	booking request; D3 (status, departure time); D1 (role)	valid request to 4.2; rejection to requester
4.2	Capture Passengers	passenger details (1 to 6)	passenger list to 4.3; rejection if more than 6
4.3	Compute Fare	passenger list; D2 (distance between stations, rate_per_km)	fare to 4.4
4.4	Allocate Seats or Waitlist	fare and passenger list; D2 (seats of the class); D4 (seats already taken, waitlist count)	allocation to 4.5; D4 (seat hold)
4.5	Generate PNR and Record Booking	allocation; payment result from 6.0; PNR enquiry; D4	payment request to 6.0; ticket (PNR); PNR status; booking event to 7.0; D4

Balance check with level 1: the level 1 process 4.0 receives booking request, passenger details, PNR enquiry, payment result and the stores D1, D2, D3, D4; it produces ticket, PNR status, payment request, booking event and writes D4. Every one of these appears exactly once at the boundary of the level 2 diagram, so the diagrams are balanced.

Entity Relationship Diagram

■ Write in lab record

```

1  +-----+ 1 places M +-----+ M for 1 +-----+
2  | User |-----| Booking |-----| Sch
3  +-----+ +-----+ +-----+
4              1| 1| 1|
5          paid by |refund | | includes
6          (1:1) |by 1:M | | (1 to 6)
7              1| M| M|
8          +-----+ | +-----+
9          | Payment | | Passenger |
10         +-----+ | +-----+
11              | | occupies
12              M| | (0..1)
13          +-----+ |
14          | Refund | |
15         +-----+ |
16              |M
17              +-----+
18
19  Extra lines to draw by hand (Station is a target of two more relationships)
20  Booking (M) ---- from_station ----> (1) Station
21  Booking (M) ---- to_station -----> (1) Station

```

Primary keys are not repeated on the diagram; underline them by hand from the data dictionary below. Every relationship, including the two Station lines, is listed here:

Relationship	Entities	Cardinality	Participation	Meaning
places	User, Booking	1:M	User partial, Booking total	A user may make many bookings; every booking belongs to exactly one user (the passenger online or the clerk at the counter)
includes	Booking, Passenger	1:M (1 to 6)	Both total	A booking lists 1 to 6 passengers; a passenger record belongs to one booking
for	Booking, Schedule	M:1	Booking total, Schedule partial	Every booking is on one train run; a schedule may have many bookings or none
run of	Schedule, Train	M:1	Schedule total, Train partial	Every schedule is one date of one train; a train may have many schedules
has stops	Train, Route, Station	M:N resolved by Route	Train total, Station partial	Route is the associative entity: one row per (train, station) with sequence, times, distance
has	Train, Coach	1:M	Both total	A train has one or more coaches; a coach belongs to one train
contains	Coach, Seat	1:M	Both total	A coach has seat_count seats; a seat belongs to one coach
priced by	Train, Fare	1:M	Both total	One fare row per (train, class); a train must have a fare for every class it carries

Relationship	Entities	Cardinality	Participation	Meaning
occupies	Passenger, Seat	M:1, optional on the Passenger side	Passenger partial, Seat partial	A confirmed passenger holds one seat for the booking's schedule; a waitlisted passenger holds none; a seat is used by many passengers over different schedules but by at most one per schedule
from_station	Booking, Station	M:1	Booking total, Station partial	Boarding station of the booking
to_station	Booking, Station	M:1	Booking total, Station partial	Alighting station of the booking
paid by	Booking, Payment	1:1	Both total once confirmed	Every completed booking has exactly one successful payment
refunded by	Booking, Refund	1:M	Booking partial, Refund total	A booking may produce several refunds (partial cancellations); every refund belongs to one booking

Data Dictionary

■ Write in lab record

Types: INT (integer), CHAR(n) (fixed length), VARCHAR(n) (variable length up to n), DATE, TIME, DATETIME, DECIMAL(p,s), ENUM (fixed set of values).

User

Attribute	Type	Size	Constraints	Description
user_id	INT	4 bytes	PK, auto-increment	Unique user identifier
name	VARCHAR	60	not null	Full name
email	VARCHAR	100	not null, unique, valid email format	Login id and notification address
mobile	CHAR	10	not null, unique, digits only	Indian mobile number for SMS
password_hash	CHAR	60	not null	bcrypt hash of the password
role	ENUM	PASSENGER, CLERK, ADMIN	not null, default PASSENGER	Access role

Passenger

Attribute	Type	Size	Constraints	Description
passenger_id	INT	4 bytes	PK, auto-increment	Unique passenger row
booking_id	CHAR	10	FK to Booking.pnr, not null	Booking this passenger travels on
name	VARCHAR	60	not null	Passenger name as on ID
age	INT	1 to 120	not null	Age in years
gender	ENUM	M, F, O	not null	Gender
berth_preference	ENUM	LOWER, MIDDLE, UPPER, SIDE_LOWER, SIDE_UPPER, NONE	default NONE	Preferred berth type
seat_id	INT	4 bytes	FK to Seat.seat_id, null when waitlisted	Allocated seat

Station

Attribute	Type	Size	Constraints	Description
station_code	VARCHAR	5	PK, 2 to 5 upper-case letters	Official station code, for example NDLS
name	VARCHAR	60	not null	Station name
city	VARCHAR	40	not null	City

Train

Attribute	Type	Size	Constraints	Description
train_no	CHAR	5	PK, 5 digits	Train number, for example 12301
name	VARCHAR	60	not null	Train name
type	ENUM	EXPRESS, SUPERFAST, PASSENGER	not null	Train category
total_coaches	INT	1 to 24	not null	Number of coaches

Route

Attribute	Type	Size	Constraints	Description
train_no	CHAR	5	PK part, FK to Train	Train
station_code	VARCHAR	5	PK part, FK to Station	Station on the route
sequence	INT	1 to 99	not null, unique per train, consecutive from 1	Order of the stop
arrival_time	TIME	HH:MM	null at first stop	Arrival at this station
departure_time	TIME	HH:MM	null at last stop	Departure from this station
distance_km	INT	0 to 5000	not null, 0 at first stop, increasing with sequence	Cumulative distance from origin

Schedule

Attribute	Type	Size	Constraints	Description
schedule_id	INT	4 bytes	PK, auto-increment	Unique run
train_no	CHAR	5	FK to Train, not null	Train that runs
run_date	DATE	YYYY-MM-DD	not null, unique with train_no, not in the past at creation	Departure date from origin
status	ENUM	SCHEDULED, CANCELLED, RESCHEDULED	not null, default SCHEDULED	Run status

Coach

Attribute	Type	Size	Constraints	Description
coach_id	INT	4 bytes	PK, auto-increment	Unique coach
train_no	CHAR	5	FK to Train, not null	Owning train
class	ENUM	SL, 3A, 2A, 1A	not null	Coach class
seat_count	INT	1 to 80	not null	Seats in this coach (SL 72, 3A 64, 2A 46, 1A 18 typical)

Seat

Attribute	Type	Size	Constraints	Description
seat_id	INT	4 bytes	PK, auto-increment	Unique seat
coach_id	INT	4 bytes	FK to Coach, not null	Owning coach
seat_no	INT	1 to 80	not null, unique within coach	Seat number printed on the ticket
berth_type	ENUM	LOWER, MIDDLE, UPPER, SIDE_LOWER, SIDE_UPPER	not null	Berth position

Booking

Attribute	Type	Size	Constraints	Description
pnr	CHAR	10	PK, 10 digits, unique, generated by the system	Passenger Name Record number
user_id	INT	4 bytes	FK to User, not null	User who made the booking
schedule_id	INT	4 bytes	FK to Schedule, not null	Train run booked
from_station	VARCHAR	5	FK to Station, not null, must precede to_station on the route	Boarding station
to_station	VARCHAR	5	FK to Station, not null	Alighting station
class	ENUM	SL, 3A, 2A, 1A	not null	Class booked
booking_time	DATETIME	YYYY-MM-DD HH:MM:SS	not null, set by the system	When payment succeeded
status	ENUM	CONFIRMED, WAITLISTED, CANCELLED	not null	Booking status
total_fare	DECIMAL	(10,2)	not null, greater than 0	Fare per passenger times passenger count, in rupees

Payment

Attribute	Type	Size	Constraints	Description
payment_id	INT	4 bytes	PK, auto-increment	Unique payment
pnr	CHAR	10	FK to Booking, not null, unique	Booking paid for
amount	DECIMAL	(10,2)	not null, equals Booking.total_fare	Amount charged
mode	ENUM	UPI, CARD, NETBANKING, CASH	not null	Payment mode; CASH only for clerk bookings
status	ENUM	INITIATED, SUCCESS, FAILED, REFUNDED	not null, default INITIATED	Payment status
txn_ref	VARCHAR	40	null for CASH, unique otherwise	Gateway transaction reference
paid_at	DATETIME	YYYY-MM-DD HH:MM:SS	null until SUCCESS	Time of successful payment

Refund

Attribute	Type	Size	Constraints	Description
refund_id	INT	4 bytes	PK, auto-increment	Unique refund
pnr	CHAR	10	FK to Booking, not null	Booking refunded
amount	DECIMAL	(10,2)	not null, 0 or more, at most Booking.total_fare	Refund amount after the time-band rule
reason	ENUM	USER_CANCEL, WAITLIST_EXPIRED, TRAIN_CANCELLED	not null	Why the refund was made
processed_at	DATETIME	YYYY-MM-DD HH:MM:SS	null until the gateway or clerk confirms	When money was returned

Fare

Attribute	Type	Size	Constraints	Description
train_no	CHAR	5	PK part, FK to Train	Train
class	ENUM	SL, 3A, 2A, 1A	PK part	Class
rate_per_km	DECIMAL	(6,2)	not null, greater than 0	Rupees per km for one passenger

Data flow dictionary

Data flow	Composition	From, to
registration	name + email + mobile + password	Passenger to 1.0
login	email + password	Passenger, Clerk, Admin to 1.0
search criteria	from_station + to_station + run_date	Passenger to 3.0
train list	0..n of (train_no + name + departure_time + arrival_time + distance_km + duration + 1..4 of (class + availability + fare))	3.0 to Passenger
availability	count of free seats, or WL + waitlist count	part of train list
booking request	schedule_id + class + from_station + to_station	Passenger, Clerk to 4.1
passenger details	1..6 of (name + age + gender + berth_preference)	Passenger, Clerk to 4.2
fare	fare_per_passenger + total_fare	4.3 to 4.4
allocation	1..6 of (passenger + (seat_id + coach_id + seat_no) or waitlist_no) + status	4.4 to 4.5
payment request	pnr + amount + mode	4.5 to 6.0, 6.0 to Payment Gateway
payment result	payment_id + status + (txn_ref) + (paid_at)	Payment Gateway to 6.0, 6.0 to 4.5
ticket (PNR)	pnr + train_no + name + run_date + from_station + to_station + class + 1..6 of (name + age + seat or WL) + total_fare + status	4.5 to Passenger, Clerk
PNR enquiry	pnr	Passenger to 4.5

Data flow	Composition	From, to
PNR status	pnr + status + 1..6 of (name + seat or WL)	4.5 to Passenger
cancellation by PNR	pnr + 1..6 of passenger_id	Passenger, Clerk to 5.0
refund details	pnr + 1..6 of (passenger_id + refund_amount) + total_refund + time_band	5.0 to Passenger, Clerk
refund request	refund_id + pnr + amount + mode	5.0 to 6.0
booking event	pnr + event (CONFIRMED, WAITLISTED, CANCELLED, PROMOTED, SCHEDULE_CHANGE) + email + mobile + details	4.0, 5.0, 2.0 to 7.0
notification request	channel (SMS, EMAIL) + recipient + message text	7.0 to Notification Service
delivery status	message_id + status (SENT, FAILED)	Notification Service to 7.0
schedule details	train_no + run_date + status	Administrator to 2.0
report request	report type (OCCUPANCY, REVENUE, CANCELLATION) + (schedule_id) + (date_from + date_to)	Administrator to 8.0

Viva Questions

✕ Do not copy. Read for understanding and the viva

- **Q:** What is the difference between a context diagram and a level 1 DFD? **A:** The context diagram shows the whole system as one process with external entities; level 1 breaks it into major processes and adds data stores.

- **Q:** Why are there no data stores on the context diagram? **A:** Stores are inside the system; the context diagram shows only the boundary.
- **Q:** What does balancing mean? **A:** The inputs and outputs of a process at one level equal the net inputs and outputs of its explosion at the next level.
- **Q:** Can a data flow go from Passenger straight to D4 Booking? **A:** No; a process must transform or record the data. Only processes connect to stores.
- **Q:** Why is Route a separate entity? **A:** Train to Station is many-to-many and the relationship carries attributes (sequence, times, distance), so it needs an associative entity.
- **Q:** What is the cardinality between Booking and Passenger? **A:** One to many, limited to 1 to 6, total participation on both sides.
- **Q:** Why is seat_id in Passenger nullable? **A:** A waitlisted passenger has no seat until promoted.
- **Q:** How does the data dictionary help the coder? **A:** It fixes the type, size, and constraint of every field before the database and code are written, so the same name means the same thing everywhere.
- **Q:** Where does the PNR get generated in the DFD? **A:** Process 4.5, after 6.0 returns a successful payment result.

Common Mistakes

✗ Do not copy. Read for understanding and the viva

- Naming processes with nouns ("Booking") or flows with verbs ("book"). Processes are verb phrases; flows are data.
- Drawing a flow between two external entities, or from an entity directly into a data store.
- Unbalanced levels: a flow appears in level 2 that never entered or left 4.0 in level 1, or the other way round.
- Forgetting cardinality and participation on the ERD, or writing M:N without resolving it.
- A data dictionary that lists only entities and not the data flows; the examiner checks that every arrow label has an entry.
- Different names for the same thing across the three models, for example "reservation" in the DFD and "booking" in the ERD.

Session Summary

■ Write in lab record

- Problem statement (a), (b), (c) copied verbatim.
- Context diagram with five external entities and its element table.
- Level 1 DFD with processes 1.0 to 8.0, tables of processes, data stores D1 to D7, and inter-process flows.
- Level 2 DFD of 4.0 Book Ticket with 4.1 to 4.5, its process table, and the balance check.
- ERD of the twelve entities with the relationship table (cardinality, participation, meaning).
- Data dictionary: one table per entity and the data flow dictionary.

 [Edit this page](#)

[< Previous](#)
Session 3

Session 5 [Next >](#)