Challenge B - Database Thinking: Bell Process Tracking

Machines Table

	Field Name	Data Type
PK	MachinePIN	Int/Varchar
	ModelName	Varchar(50)

Bays Table

	Field Name	Data Type
PK	BayID	Int/Varchar
	Bay Name	Varchar(50)
FK	SupervisorID	References supervisor table primary key

Supervisors Table

	Field Name	Data Type
PK	SupervisorID	Int/Varchar
	SupervisorName	Varchar(100)

MachineMovements Table

	Field Name	Data Type
PK	MovementID	Int
FK	MachinePIN	References Machines Table Primary key
FK	BayID	References Bays table primary key
	DateTimeEntered	DateTime
FK	ScannerID	Varchar(100)

Employees

	Field Name	Data Type
PK	EmployeeID	Int/Varchar
	EmployeeName	Varchar(100)

Why my design avoids duplication and supports scalability.

My design avoids duplication due to normalising it into its third normal form. Separating references into their own independent tables reduces the need to replicate data, for example: In the MachineMovements Table there is foreign keys linking to the machines table, bay table and employees table, this prevents duplication as there will only be one record corresponding to the ID. If the employee name was stored in the MachineMovements Table then if you want to change his name you would have to change all the records with his name in it, with this design the employees name is only one row, with no duplication. It is scalable due to its current structure making it faster for the database to search, index, and join tables using the primary and foreign keys. Its design with multiple tables also makes it easier to make changes to a specific table without needing to make big changes.

SQL Create Query for entire all tables

```
CREATE TABLE Supervisors (
    SupervisorID VARCHAR(50) PRIMARY KEY,
    SupervisorName VARCHAR(100)
);
```

```sql
CREATE TABLE Employees (
    EmployeeID VARCHAR(100) PRIMARY KEY,
    EmployeeName VARCHAR(100)
);

CREATE TABLE Machines (
    MachinePIN INT PRIMARY KEY,
    ModelName VARCHAR(50)
);

CREATE TABLE Bays (
    BayID VARCHAR(50) PRIMARY KEY,
    BayName VARCHAR(50),
    SupervisorID VARCHAR(50),
    FOREIGN KEY (SupervisorID) REFERENCES
Supervisors(SupervisorID)
);

CREATE TABLE MachineMovements (
    MovementID INT PRIMARY KEY,
    MachinePIN INT NOT NULL,
    BayID VARCHAR(50) NOT NULL,
    DateTimeEntered DATETIME NOT NULL,
    ScannerID VARCHAR(100) NOT NULL,
    FOREIGN KEY (MachinePIN) REFERENCES
Machines(MachinePIN),
    FOREIGN KEY (BayID) REFERENCES Bays(BayID),
    FOREIGN KEY (ScannerID) REFERENCES
Employees(EmployeeID)
```

```
);
```

SQL Query to show latest bay for each machine

```
SELECT mm.MachinePIN, mm.BayID, mm.DateTimeEntered
FROM MachineMovements mm
INNER JOIN (
SELECT MachinePIN, MAX(DateTimeEntered) AS
LatestTime FROM MachineMovements GROUP BY MachinePIN
)
latest ON mm.MachinePIN = latest.MachinePIN AND
mm.DateTimeEntered = latest.LatestTime;
```

Example output

Example data in the MachineMovements Table

MovementID	MachinePIN	BayID	DateTimeEntered	ScannerID
1	1	BayA	2025-10-26 08:00:00	Emp1
2	2	BayB	2025-10-26 09:15:00	Emp2
3	1	BayC	2025-10-27 14:30:00	Emp1
4	3	BayA	2025-10-27 10:00:00	Emp3
5	2	BayA	2025-10-28 11:45:00	Emp2

Example Output

MachinePIN	BayID	DateTimeEntered
1	BayC	2025-10-27 14:30:00
2	BayA	2025-10-27 10:00:00
3	BayA	2025-10-28 11:45:00