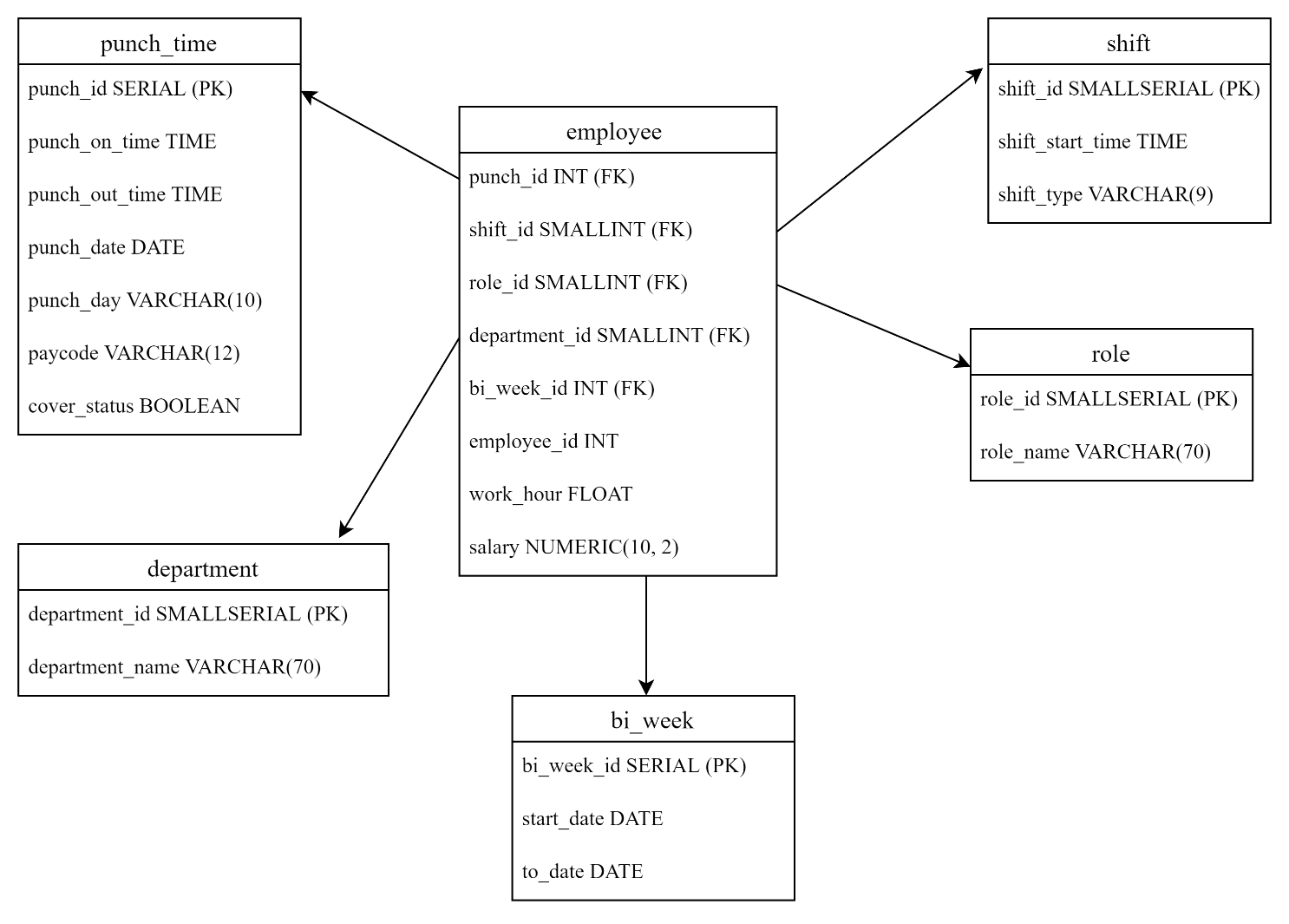
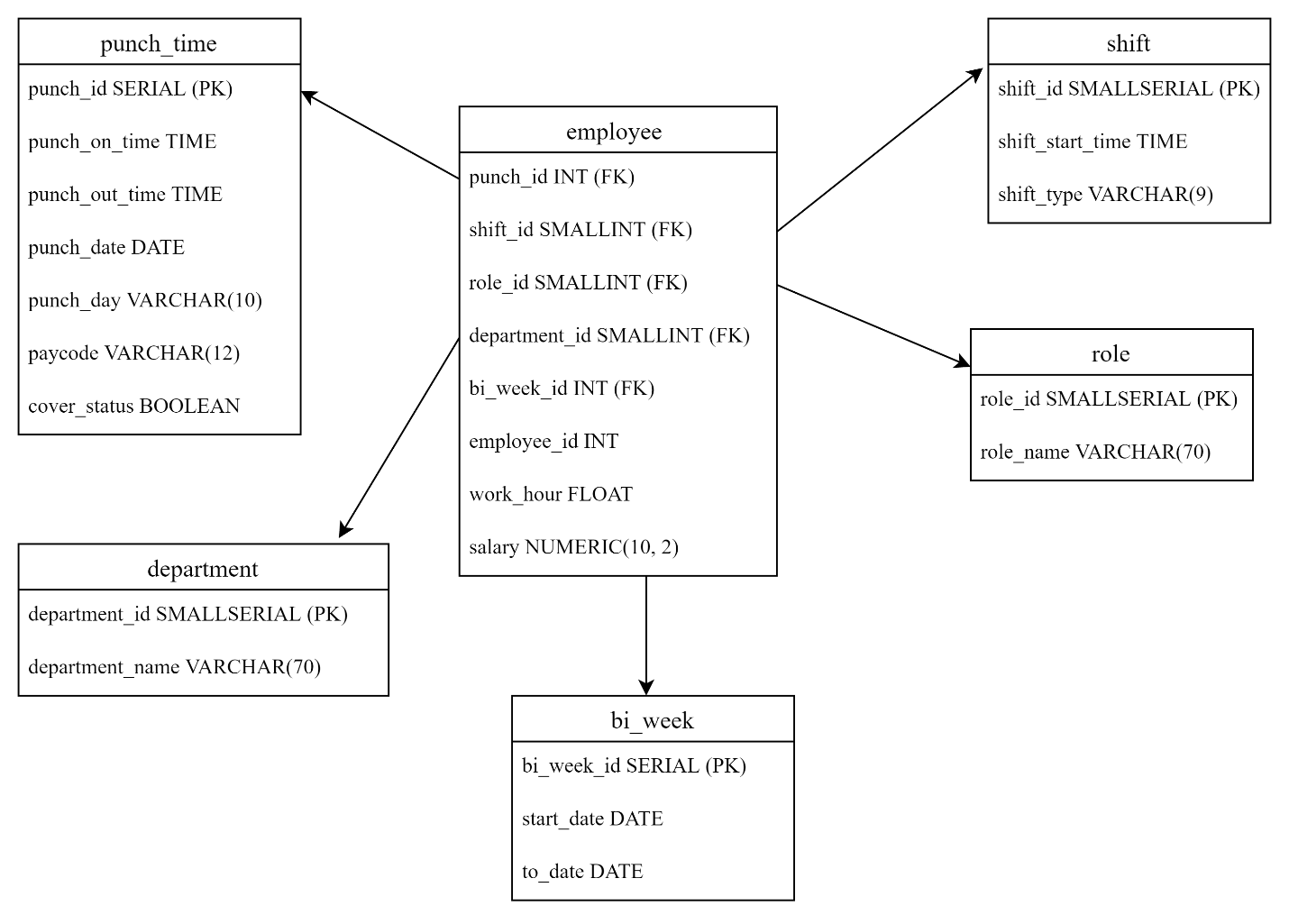
**Entity Identification**

1. punch\_time 🡪 (dimension table)
2. shift 🡪 (dimension table)
3. role 🡪 (dimension table)
4. department 🡪 (dimension table)
5. bi\_week 🡪 (dimension table)
6. employee 🡪 (fact table)

**Logical Design**

|  |  |  |
| --- | --- | --- |
| Entity | Attributes | description |
| punch\_time | punch\_id (PK)  punch\_on\_time  punch\_out\_time  punch\_date  punch\_day  pay\_code  cover\_status | auto generated primary key  time type data  time type data  date type data  shows the weekdays  shows the paycode of employee  shows if employee need to cover (BOOLEAN) |
| shift | shift\_id (PK)  shift\_start\_time  shift\_type | auto generated primary key  punch\_on\_time  type of shift based on punch\_on\_time |
| role | role\_id (PK)  role\_name | auto generated primary key  shows the name of role of employee |
| department | department\_id (PK)  department\_name | auto generated primary key  shows the department name |
| bi\_week | bi\_week\_id (PK)  start\_date  to\_date | auto generated primary key  shows the start date for starting biweekly analysis  shows to which date do the biweekly analysis is |
| employee | punch\_id (FK)  shift\_id (FK)  department\_id (FK)  bi\_week\_id (FK)  employee\_id  work\_hour  salary | valid id from punch\_time table  valid id from shift table  valid id from department table  valid id from bi\_week table  shows the employee id of the employee  shows the working hour of the employee  shows the salary of the employee |



**Physical model**

[**Diagram link**](https://drive.google.com/file/d/1UbG9wT3NE1uVtc_kQ3KX5xsqVqaRd7-W/view?usp=sharing)

**Physical implementation**

CREATE SCHEMA IF NOT EXISTS etl;

CREATE TABLE IF NOT EXISTS etl.punch\_time(

punch\_id SERIAL PRIMARY KEY,

punch\_on\_time TIME,

punch\_out\_time TIME,

punch\_date DATE,

punch\_day VARCHAR(10),

pay\_code VARCHAR(12) NOT NULL,

cover\_status BOOLEAN NOT NULL

);

CREATE TABLE IF NOT EXISTS etl.department(

department\_id SMALLSERIAL PRIMARY KEY,

department\_name VARCHAR(70) NOT NULL

);

CREATE TABLE IF NOT EXISTS etl.shift(

shift\_id SMALLSERIAL PRIMARY KEY,

shift\_start\_time TIME,

shift\_type VARCHAR(9)

);

CREATE TABLE IF NOT EXISTS etl.role(

role\_id SMALLSERIAL PRIMARY KEY,

role\_nane VARCHAR(70)

);

CREATE TABLE IF NOT EXISTS etl.bi\_week(

bi\_week\_id SERIAL PRIMARY KEY,

start\_date DATE,

to\_date DATE

);

CREATE TABLE IF NOT EXISTS etl.employee(

punch\_id INT,

shift\_id SMALLINT,

role\_id SMALLINT,

department\_id SMALLINT,

bi\_week\_id INT,

employee\_id INT,

work\_hour FLOAT,

salary NUMERIC(10, 2),

FOREIGN KEY (punch\_id) REFERENCES etl.punch\_time(punch\_id),

FOREIGN KEY (shift\_id) REFERENCES etl.shift(shift\_id),

FOREIGN KEY (role\_id) REFERENCES etl.role(role\_id),

FOREIGN KEY (department\_id) REFERENCES etl.department(department\_id),

FOREIGN KEY (bi\_week\_id) REFERENCES etl.bi\_week(bi\_week\_id)

);