Solutions

Task 1.1

```
CREATE TABLE employees (
emp_id SERIAL PRIMARY KEY,
first_name TEXT NOT NULL,
last_name TEXT NOT NULL,
job_position TEXT NOT NULL,
salary decimal(8,2),
start_date DATE NOT NULL,
birth_date DATE NOT NULL,
store_id INT REFERENCES store(store_id),
department_id INT,
manager_id INT
);
```

Task 1.2

```
CREATE TABLE departments (
department_id SERIAL PRIMARY KEY,
department TEXT NOT NULL,
division TEXT NOT NULL);
```

Task 2

```
ALTER TABLE employees
ALTER COLUMN department_id SET NOT NULL,
ALTER COLUMN start_date SET DEFAULT CURRENT_DATE,
ADD COLUMN end_date DATE,
ADD CONSTRAINT birth_check CHECK(birth_date < CURRENT_DATE);
ALTER TABLE employees
RENAME job_position TO position_title;
```

Task 3.1

```
INSERT INTO employees
VALUES
(1,'Morrie','Conaboy','CTO',21268.94,'2005-04-30','1983-07-10',1,1,NULL,NULL),
(2,'Miller','McQuarter','Head of BI',14614.00,'2019-07-23','1978-11-09',1,1,1,NULL),
(3,'Christalle','McKenny','Head of Sales',12587.00,'1999-02-05','1973-01-09',2,3,1,NULL),
(4,'Sumner','Seares','SQL Analyst',9515.00,'2006-05-31','1976-08-03',2,1,6,NULL),
(5,'Romain','Hacard','BI Consultant',7107.00,'2012-09-24','1984-07-14',1,1,6,NULL),
(6,'Ely','Luscombe','Team Lead Analytics',12564.00,'2002-06-12','1974-08-01',1,1,2,NULL),
(7,'Clywd','Filyashin','Senior SQL Analyst',10510.00,'2010-04-05','1989-07-23',2,1,2,NULL),
(8,'Christopher','Blague','SQL Analyst',9428.00,'2007-09-30','1990-12-07',2,2,6,NULL),
(9,'Roddie','Izen','Software Engineer',4937.00,'2019-03-22','2008-08-30',1,4,6,NULL),
(10,'Ammamaria','Izhak','Customer Support',2355.00,'2005-03-17','1974-07-27',2,5,3,'2013-04-14'),
(11,'Carlyn','Stripp','Customer Support',3060.00,'2013-09-06','1981-09-05',1,5,3,NULL),
(12,'Reuben','McRorie','Software Engineer',7119.00,'1995-12-31','1958-08-15',1,5,6,NULL),
(13,'Gates','Raison','Marketing Specialist',3910.00,'2013-07-18','1986-06-24',1,3,3,NULL),
(14,'Jordanna','Raitt','Marketing Specialist',5844.00,'2011-10-23','1993-03-16',2,3,3,NULL),
(15,'Guendolen','Motton','BI Consultant',8330.00,'2011-01-10','1980-10-22',2,3,6,NULL),
(16,'Doria','Turbat','Senior SQL Analyst',9278.00,'2010-08-15','1983-01-11',1,1,6,NULL),
(17,'Cort','Bewlie','Project Manager',5463.00,'2013-05-26','1986-10-05',1,5,3,NULL),
(18,'Margarita','Eaden','SQL Analyst',5977.00,'2014-09-24','1978-10-08',2,1,6,'2020-03-16'),
(19,'Hetty','Kingaby','SQL Analyst',7541.00,'2009-08-17','1999-04-25',1,2,6,NULL),
(20,'Lief','Robardley','SQL Analyst',8981.00,'2002-10-23','1971-01-25',2,3,6,'2016-07-01'),
(21,'Zaneta','Carlozzi','Working Student',1525.00,'2006-08-29','1995-04-16',1,3,6,'2012-02-19'),
(22,'Giana','Matz','Working Student',1036.00,'2016-03-18','1987-09-25',1,3,6,NULL),
(23,'Hamil','Evershed','Web Developper',3088.00,'2022-02-03','2012-03-30',1,4,2,NULL),
(24,'Lowe','Diamant','Web Developper',6418.00,'2018-12-31','2002-09-07',1,4,2,NULL),
(25,'Jack','Franklin','SQL Analyst',6771.00,'2013-05-18','2005-10-04',1,2,2,NULL),
(26,'Jessica','Brown','SQL Analyst',8566.00,'2003-10-23','1965-01-29',1,1,2,NULL);
```

Task 3.2

```
INSERT INTO departments
VALUES (1, 'Analytics','IT'),
(2, 'Finance','Administration'),
(3, 'Sales','Sales'),
(4, 'Website','IT'),
(5, 'Back Office','Administration')
```

Task 4.1

```
UPDATE employees
SET position_title = 'Senior SQL Analyst'
WHERE emp_id=25;
UPDATE employees
SET salary=7200
WHERE emp_id=25;
```

Task 4.2

```
UPDATE employees
SET position_title='Customer Specialist'
WHERE position_title='Customer Support';
```

Task 4.3

```
UPDATE employees
SET salary=salary*1.06
WHERE position_title LIKE '%SQL Analyst';
```

Task 4.4

```
SELECT ROUND(AVG(salary),2) FROM employees
WHERE position_title='SQL Analyst'
```

Task 5

```
SELECT
emp.*,
CASE WHEN emp.end_date IS NULL THEN 'true'
ELSE 'false'
END as is_active,
mng.first_name ||' '|| mng.last_name AS manager_name
FROM employees emp
LEFT JOIN employees mng
ON emp.manager_id=mng.emp_id;
```

Task 5.2

```
CREATE VIEW v_employees_info
AS
SELECT
emp.*,
CASE WHEN emp.end_date IS NULL THEN 'true'
ELSE 'false'
END as is_active,
mng.first_name ||' '|| mng.last_name AS manager_name
FROM employees emp
LEFT JOIN employees mng
ON emp.manager_id=mng.emp_id;
```

Task 6

```
SELECT
position_title,
ROUND(AVG(salary),2)
FROM v_employees_info
GROUP BY position_title
ORDER BY 2;
```

Task 7

```
SELECT
division,
ROUND(AVG(salary),2)
FROM employees e
LEFT JOIN departments d
ON e.department_id=d.department_id
GROUP BY division
ORDER BY 2
```

Task 8.1

```
SELECT
emp_id,
first_name,
last_name,
position_title,
salary,
ROUND(AVG(salary) OVER(PARTITION BY position_title),2) as avg_position_sal
FROM employees
ORDER BY 1;
```

Task 8.2

```
SELECT
COUNT(*)
FROM (
SELECT
emp_id,
salary,
ROUND(AVG(salary) OVER(PARTITION BY position_title),2) as avg_pos_sal
FROM employees) a
WHERE salary<avg_pos_sal;
```

Task 9:

```
SELECT
emp_id,
salary,
start_date,
SUM(salary) OVER(ORDER BY start_date) as avg_pos_sal
FROM employees;
```

Task 10:

```
SELECT
start_date,
SUM(salary) OVER(ORDER BY start_date)
FROM (
SELECT
emp_id,
salary,
start_date
FROM employees
UNION
SELECT
emp_id,
-salary,
end_date
FROM v_employees_info
WHERE is_active ='false'
ORDER BY start_date) a
```

Task 11.1

```
SELECT
first_name,
position_title,
salary
FROM employees e1
WHERE salary = (SELECT MAX(salary)
                FROM employees e2
                WHERE e1.position_title=e2.position_title)
```

Task 11.2

```
SELECT
first_name,
position_title,
salary,
(SELECT ROUND(AVG(salary),2) as avg_in_pos FROM employees e3
WHERE e1.position_title=e3.position_title)
FROM employees e1
WHERE salary = (SELECT MAX(salary)
                FROM employees e2
                WHERE e1.position_title=e2.position_title)
```

Task 11.3

```
SELECT
```

```
first_name,
position_title,
salary,
(SELECT ROUND(AVG(salary),2) as avg_in_pos FROM employees e3
WHERE e1.position_title=e3.position_title)
FROM employees e1
WHERE salary = (SELECT MAX(salary)
                     FROM employees e2
                     WHERE e1.position_title=e2.position_title)
AND salary<>(SELECT ROUND(AVG(salary),2) as avg_in_pos FROM employees e3
WHERE e1.position_title=e3.position_title)
```

Task 12

```
SELECT
division,
department,
position_title,
SUM(salary),
COUNT(*),
ROUND(AVG(salary),2)
FROM employees
NATURAL JOIN departments
GROUP BY
ROLLUP(
division,
department,
position_title
)
ORDER BY 1,2,3
```

Task 13

```
SELECT
emp_id,
position_title,
department,
salary,
RANK() OVER(PARTITION BY department ORDER BY salary DESC)
FROM employees
NATURAL LEFT JOIN departments
```

Task 14

```
SELECT * FROM
```

```
(
SELECT
emp_id,
position_title,
department,
salary,
RANK() OVER(PARTITION BY department ORDER BY salary DESC)
FROM employees
NATURAL LEFT JOIN departments) a
WHERE rank=1
```