Assignment 6

Question 1)

```sql
-- Shayan 27027
/*
Find the employee with the 9th Highest salary from the employees table
(Among row_num, rank, and dense_rank, use the function that gives the 9th highest salary
most accurately, even when there are duplicates). Don't use any of the subqueries, use a CTE.
*/
with ninth_highsalary as(
select employee_id , salary,
       DENSE_RANK() over(order by salary desc) as dense_rank
from OEHR_EMPLOYEES
)

select *
from ninth_highsalary
where dense_rank=9
```

Messages

employee_id	salary	dense_rank
162	10500	9
149	10500	9

Question 2)

```sql
-- Shayan 27027
/*
 Use a CTE to calculate how many people have salary above or below the company average.
 I want two columns, with two rows, one column that shows “Above Average” and “Below Average”, other
 column that shows the respective count.
*/

with company_average as (
    select salary, avg(salary) over() as avg --created whole column for salary
    from OEHR_EMPLOYEES
),
salary_type as (
    select case
            when salary > avg  then 'Above Average'
            when salary < avg then 'Below Average'
            else 'Average'  -- just included though this case isn't there in the data
           end as salary_category
    from company_average
)
select
      salary_category,
      count(*) as count
from salary_type
group by salary_category;
```

Results Messages

salary_category	count
Above Average	51
Below Average	56

Question 3)

```sql
-- Shayan 27027
/*
 Create a CTE to only have those employees who have salary above company average, and then join this CTE
 with the department table to bring out the count of employees
 (only those employees with salary above company average) with department names.
*/

-- finding above-average salaried employees
with average as(
      select * , avg(salary) over() as avg
      from OEHR_EMPLOYEES
 ),
 abv_avg as(
      select a.*
      from OEHR_EMPLOYEES as a
      inner join average as b
      on a.EMPLOYEE_ID=b.EMPLOYEE_ID
      where a.salary > b.avg
 )
 -- above average salaried employees count for each department
 select b.department_name , count(*) as count
 from abv_avg as a
 left join OEHR_DEPARTMENTS as b
 on a.DEPARTMENT_ID=b.DEPARTMENT_ID
 group by b.DEPARTMENT_NAME
 order by count desc
```

100 %

Results | Messages

	department_name	count
1	Sales	30
2	Finance	6
3	Shipping	4
4	Executive	3
5	Accounting	2

Query executed successfully. DESKTOP-S0NQ4VV\SQLEXPRESS ... DESKTOP-S0NQ4VV\Tesla ... HR_DATA 00:00:00 11 rows

Question 4)

```sql
-- Shayan 27027
/*
 Give me all the employees who have the lowest salaries in their respective departments.
 Use Dense rank.
*/

with employeedepart_ranking as(
    select * ,
        DENSE_RANK() over(partition by department_id order by salary ) as Ranking
    from OEHR_EMPLOYEES
)

select EMPLOYEE_ID, FIRST_NAME,LAST_NAME,DEPARTMENT_ID,SALARY
from employeedepart_ranking
where Ranking=1   -- department 90 has 2 lowest salaries
```

%

Results | Messages

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID	SALARY
132	TJ	Olson	50	2100
107	Diana	Lorentz	60	4200
204	Hermann	Baer	70	10000
173	Sundita	Kumar	80	6100
101	Neena	Kochhar	90	17000
102	Lex	De Haan	90	17000
113	Luis	Popp	100	6900
206	William	Gietz	110	8300

Query executed successfully. DESKTOP-S0NQ4VV\SQLEXPRESS ... DESKTOP-S0NQ4VV\Tesla ... HR_DATA 00:00:00 13 rows

Question 5)

```sql
-- Shayan 27027
/*
Give me all the employees who have the Highest salaries in their respective departments and job titles.
Use Dense rank.
*/

with high_ranking as(
   select EMPLOYEE_ID, concat(FIRST_NAME, ' ', LAST_NAME) as name, a.department_id, b.department_name, salary,
       DENSE_RANK() over(partition by a.department_id, b.department_name order by salary desc ) as Ranking
   from OEHR_EMPLOYEES as a
   left join OEHR_DEPARTMENTS as b
   on a.DEPARTMENT_ID=b.DEPARTMENT_ID
)

Select *
from high_ranking
where Ranking=1
```

124 %

Results | Messages

	EMPLOYEE_ID	name	department_id	department_name	salary	Ranking
1	178	Kimberely Grant	NULL	NULL	7000	1
2	200	Jennifer Whalen	10	Administration	4400	1
3	201	Michael Hartstein	20	Marketing	13000	1
4	114	Den Raphaely	30	Purchasing	11000	1
5	203	Susan Mavris	40	Human Resources	6500	1
6	121	Adam Fripp	50	Shipping	8200	1
7	103	Alexander Hunold	60	IT	9000	1
8	204	Hermann Baer	70	Public Relations	10000	1
9	145	John Russell	80	Sales	14000	1

Query executed successfully. DESKTOP-S0NQ4VV\SQLEXPRESS ... DESKTOP-S0NQ4VV\Tesla ... HR_DATA 00:00:00 12 rows

Question 6)

```sql
--Shayan 27027
/*
 Use Row_num, rank, and dense_rank together in one query on the employees table to rank employees with respect to salary.
 Along with displaying the starting 10 rows, make sure to write the difference between these three functions.
*/
select top 10 *,
      ROW_NUMBER() over (order by salary) as row_num,
      rank() over (order by salary) as rank,
      dense_rank() over (order by salary) as dense_rank
from OEHR_EMPLOYEES;
-- Differnce --
/* Row_number -->  just numbers the rows uniquely based on salary and duplicates aren't catered */
/* Rank --> ranks the salaries and the duplicates gets the same rank but it skips the rank due duplicates */
/* Dense_rank --> ranks the salaries and duplicates gets the same rank but it doesn't skips ranks */
```

%

Results | Messages

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	row_num	rank	dense_rank
132	TJ	Olson	TJOLSON	650.124.8234	2019-09-30	ST_CLERK	2100	NULL	121	50	1	1	1
128	Steven	Markle	SMARKLE	650.124.1434	2020-08-28	ST_CLERK	2200	NULL	120	50	2	2	2
136	Hazel	Philtanker	HPHILTAN	650.127.1634	2020-07-28	ST_CLERK	2200	NULL	122	50	3	2	2
135	Ki	Gee	KGEE	650.127.1734	2020-06-02	ST_CLERK	2400	NULL	122	50	4	4	3
127	James	Landry	JLANDRY	650.124.1334	2019-07-06	ST_CLERK	2400	NULL	120	50	5	4	3
119	Karen	Colmenares	KCOLMENA	515.127.4566	2020-01-30	PU_CLERK	2500	NULL	114	30	6	6	4
140	Joshua	Patel	JPATEL	650.121.1834	2018-09-26	ST_CLERK	2500	NULL	123	50	7	6	4
131	James	Marlow	JAMRLOW	650.124.7234	2017-08-08	ST_CLERK	2500	NULL	121	50	8	6	4
144	Peter	Vargas	PVARGAS	650.121.2004	2018-12-29	ST_CLERK	2500	NULL	124	50	9	6	4

Query executed successfully. DESKTOP-S0NQ4VV\SQLEXPRESS ... DESKTOP-S0NQ4VV\Tesla ... HR_DATA 00:00:00 10 rows

Question 7)

```sql
-- Shayan 27027
/*
List the employees who have a job title that is not present in the job history table.
Use CTE, not subquery.
*/

with Job as(
    SELECT Distinct JOB_ID
    FROM OEHR_JOB_HISTORY
)
SELECT a.employee_id, a.job_id
FROM OEHR_EMPLOYEES a
LEFT JOIN Job b
ON a.job_id = b.job_id
WHERE b.job_id IS NULL;

-- Shayan 27027
```

24 %

Results | Messages

	employee_id	job_id
1	100	AD_PRES
2	101	AD_VP
3	102	AD_VP
4	108	FI_MGR
5	109	FI_ACCOUNT
6	110	FI_ACCOUNT
7	111	FI_ACCOUNT
8	112	FI_ACCOUNT
9	113	FI_ACCOUNT

Query executed successfully. DESKTOP-S0NQ4VV\SQLEXPRESS ... DESKTOP-S0NQ4VV\Tesla ... HR_DATA 00:00:00 43 rows

Question 8)

```sql
-- Shayan 27027
/*
Find the names of employees who have been in the company longer than the average tenure of employees
in their department. Use CTE, not subquery.
*/
with average as (
   SELECT DEPARTMENT_ID,
      AVG(DATEDIFF(Day, HIRE_DATE, GETDATE())) AS AvgTenure
   FROM OEHR_EMPLOYEES
   GROUP BY DEPARTMENT_ID
)

select first_name, last_name, a.department_id, datediff(day, HIRE_DATE,getdate()) as diff, AvgTenure
from OEHR_EMPLOYEES as a
left join average as b
on a.DEPARTMENT_ID=b.DEPARTMENT_ID
where datediff(day, HIRE_DATE,getdate())  > AvgTenure
```

4 %

Results | Messages

first_name	last_name	department_id	diff	AvgTenure
Michael	Hartstein	20	3002	2728
Den	Raphaely	30	3439	2542
Alexander	Khoo	30	3277	2542
Winston	Taylor	50	2295	2270
Nandita	Sarchand	50	3023	2270
Alexis	Bull	50	2633	2270
Kelly	Chung	50	2519	2270
Jennifer	Dilly	50	2459	2270
Sarah	Bell	50	3015	2270
Britney	Everett	50	2622	2270
Matthew	Weiss	50	2850	2270

Query executed successfully. DESKTOP-S0NQ4VV\SQLEXPRESS ... DESKTOP-S0NQ4VV\Tesla ... HR_DATA 00:00:00 50 rows

Question 9)

```sql
-- Shayan 27027
/*
 Along with all the employee data, show me the difference in salary of the current employee and
 employee who was hired just before current employee. Use only employee table.
*/

with prev_salary as(
   select *, lag(salary) over(order by hire_date) as previous
   from OEHR_EMPLOYEES
)

Select *, abs((salary-previous)) as Difference
from prev_salary
```

9 %

Results | Messages

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	previous	Difference
100	Steven	King	SKING	515.123.4567	2007-12-07	AD_PRES	24000	NULL	NULL	90	NULL	NULL
200	Jennifer	Whalen	JWHALEN	515.123.4444	2008-03-08	AD_ASST	4400	NULL	101	10	24000	19600
101	Neena	Kochhar	NKOCHHAR	515.123.4568	2010-03-13	AD_VP	17000	NULL	100	90	4400	12600
103	Alexander	Hunold	AHUNOLD	590.423.4567	2010-06-25	IT_PROG	9000	NULL	102	60	17000	8000
104	Bruce	Ernst	BERNST	590.423.4568	2011-11-10	IT_PROG	6000	NULL	103	60	9000	3000
102	Lex	De Haan	LDEHAAN	515.123.4569	2013-07-05	AD_VP	17000	NULL	100	90	6000	11000

Query executed successfully. DESKTOP-S0NQ4VV\SQLEXPRESS ... DESKTOP-S0NQ4VV\Tesla ... HR_DATA 00:00:00 107 rows