# *Database Management I (420-D10-HR)*

# *Happy Valley Kennels Project*

# *Assignment 7 – Oracle Queries*

Date assigned: Wednesday, November 9, 2016

Date due: **Friday, December 2, 2016, 11:00pm**

This assignment is tough. Be sure to take advantage of lab sessions with the professor to work on this. It will be difficult to get this on your own.

**Learning Objectives**

After completing this assignment, the student will be able to

* Create SQL queries to retrieve information from an Oracle database
* Create views using Oracle SQL

**To be uploaded to Moodle:**

1. Rename this file to ***username***\_**D10\_A07\_Queries.docx** file containing your responses.
2. You must include for each query:
   1. Your SQL
   2. Sample output from your query

**To do:**

* + 1. Write SQL statements for the queries, views and inserts described on the next page.
* Appropriate column headings should be given for reports.
* The output should be formatted appropriately (e.g. money should include a $ and only two decimal places)

|  |  |
| --- | --- |
| **Query** | **Mark** |
| 1. Missing or expired vaccinations | 25 |
| 1. Most days stayed | 20 |
| 1. hvk\_reservation\_charges\_vw View | 25 |
| 1. invoices | 15 |
| 1. Organization | 5 |
| **Total** | **90** |

**Happy Valley Kennels**

**Queries, Views and Updates**

1. Write a query listing all pets with reservations that start on or after **Nov. 10, 2016** who are missing vaccinations or whose vaccinations will have expired before the end of the reservation. Your query should include the owner name and phone number, the pet’s name and type (dog or cat), the reservation start and end dates, the vaccination name and whether the vaccination is missing or expired. For expired vaccinations include the expiry date. See the sample query output in the appendix. Verify that your test cases from assignment 6 all work. Sort by Owner and Reservation start date. (Sample output shown in Assignment #6)

Pet: 5 Vacc: 7

Pet: 7 Vacc: 3,5

Pet: 14 Vacc: 6

Pet 16 vacc 2,5

Pet 20 vacc 3,4

Pet 37 vacc 10

Pet 38 vacc 9

1. Write a query to select the pet(s) who has/have stayed the most number of days at HVK in the past. Only reservations whose end date is before today should be considered. The output should include the pet name, type, total number of days stayed in the past, the owner’s name and phone number.

Sample output:

**Pet(s) who has(have) stayed the most number of days in the past**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pet Name** | **Pet Type** | **Days Stayed** | **Customer** | **Phone** |
| Huxley | Dog | 93 | Mike O'Phone | (819)555-2222 |

/\*Question 2\*/

SELECT DISTINCT p.pet\_name AS "Pet Name", 'Dog' AS "Pet Type",

SUM(r.reservation\_end\_date - r.reservation\_start\_date) AS "Days Stayed",

o.owner\_first\_name||' '||o.owner\_last\_name AS "Customer",

'('||SUBSTR(o.owner\_phone, 0, 3)||')'||SUBSTR(o.owner\_phone, 4, 3)||'-'||SUBSTR(o.owner\_phone, 7) AS "Phone"

FROM hvk\_reservation r, hvk\_pet\_reservation pr, hvk\_pet p, hvk\_owner o, hvk\_dog d

WHERE r.reservation\_start\_date <= SYSDATE

AND pr.res\_reservation\_number = r.reservation\_number

AND pr.pet\_pet\_number = p.pet\_number

AND o.owner\_number = p.own\_owner\_number

AND p.pet\_number = d.pet\_pet\_number

GROUP BY p.pet\_name, o.owner\_last\_name, o.owner\_first\_name, o.owner\_phone

HAVING SUM(r.reservation\_end\_date - r.reservation\_start\_date) >= ALL(

SELECT DISTINCT SUM(r.reservation\_end\_date - r.reservation\_start\_date) AS "Days Stayed"

FROM hvk\_reservation r, hvk\_pet\_reservation pr, hvk\_pet p, hvk\_owner o

WHERE r.reservation\_start\_date <= SYSDATE

AND pr.res\_reservation\_number = r.reservation\_number

AND pr.pet\_pet\_number = p.pet\_number

AND o.owner\_number = p.own\_owner\_number

GROUP BY p.pet\_name, 'Customer', 'Phone', o.owner\_first\_name, o.owner\_phone

)

Union

SELECT DISTINCT p.pet\_name AS "Pet Name", 'Cat' AS "Pet Type",

SUM(r.reservation\_end\_date - r.reservation\_start\_date) AS "Days Stayed",

o.owner\_first\_name||' '||o.owner\_last\_name AS "Customer",

'('||SUBSTR(o.owner\_phone, 0, 3)||')'||SUBSTR(o.owner\_phone, 4, 3)||'-'||SUBSTR(o.owner\_phone, 7) AS "Phone"

FROM hvk\_reservation r, hvk\_pet\_reservation pr, hvk\_pet p, hvk\_owner o, hvk\_cat c

WHERE r.reservation\_start\_date <= SYSDATE

AND pr.res\_reservation\_number = r.reservation\_number

AND pr.pet\_pet\_number = p.pet\_number

AND o.owner\_number = p.own\_owner\_number

AND p.pet\_number = c.pet\_pet\_number

GROUP BY p.pet\_name, o.owner\_first\_name, o.owner\_last\_name, o.owner\_phone

HAVING SUM(r.reservation\_end\_date - r.reservation\_start\_date) >= ALL(

SELECT DISTINCT SUM(r.reservation\_end\_date - r.reservation\_start\_date) AS "Days Stayed"

FROM hvk\_reservation r, hvk\_pet\_reservation pr, hvk\_pet p, hvk\_owner o

WHERE r.reservation\_start\_date <= SYSDATE

AND pr.res\_reservation\_number = r.reservation\_number

AND pr.pet\_pet\_number = p.pet\_number

AND o.owner\_number = p.own\_owner\_number

GROUP BY p.pet\_name, 'Customer', 'Phone', o.owner\_first\_name, o.owner\_phone

)

;

1. Write a view called **hvk\_reservation\_charges\_vw** that lists all the charges for all the reservations. Your view should include the reservation number, the customer number, the pet type, the pet’s number and name, the service description, the run number or cat suite (for boarding only), the reservation start and end dates, the service frequency and the daily rate for the service. The service frequency is for is the frequency for special services if the frequency is non-null. For all other charges, the frequency is the number of days for the reservation. See the sample view contents in the appendix. Use your test cases from assignment 6 to test your view. (**Note**: do not format the date or currency amounts using TO\_CHAR for the view.). (Sample output shown in Assignment #6)
2. Create a query to list the invoice total for each reservation. Use the view you created in the previous question. Include the reservation number, owner name, start date, end date, the number of days for the reservation and the invoice total. Do not subtract discount amounts.