

## 1. PARTE UNO. Refactorización

### A. Modelo conceptual

El único cambio fue añadir la relación de “has\_manager” entre los conceptos “Shift” y “Staff” la cual había sido incluida en el modelo lógico pero no en el conceptual.

### B. Modelo lógico

Cambiamos varias cosas: se señaló una llave foránea que se había olvidado del laboratorio pasado (id del contacto de las compañías); se introdujo el nombre de la compañía como llave única; se identificaron los atributos que pueden ser nulos.

Para decidir cuáles atributos podrían ser nulos revisamos las tablas de la base de datos usando el comando DESCRIBE. Luego, de aquellos atributos que pueden ser nulos según la base de datos, se decidió que para la tabla “Shift\_type” no hay razón para que los atributos “Start\_time” y “End\_time” sean nulos, ya que, si se asigna un turno a un empleado sin estos atributos, el empleado no sabría a qué hora empezar o terminar de trabajar.

## 2. PARTE DOS. División por ciclos

A. Respecto a los tipos de los atributos “Manager”, “Operator”, “Engineer” en la tabla “Level”, estos podrían ser booleanos ya que solo toman dos valores, “Y” y “NULL”, y corresponden principalmente a tener o no una característica. Sin embargo, como es ingeniería reversa sobre una base de datos ya existente, se mantienen los tipos de datos originales.

## 3. PARTE TRES. Ciclo uno

### C) Modelo conceptual. Consultas Operativas. (¿qué ofrece?)

1. Implemente las siguientes nuevas consultas
  - a. Los empleados que más turnos han trabajado

```
SELECT s.First_name, s.Last_name, l.Level_code, COUNT(*) AS "no_shifts"
FROM Shift sh
JOIN Staff s
ON s.Staff_code IN (sh.Manager, sh.Operator, sh.Engineer1, sh.Engineer2)
JOIN Level l
ON s.Level_code = l.Level_code
GROUP BY s.First_name, s.Last_name, l.Level_code
ORDER BY no_shifts DESC;
```

First_name	Last_name	Level_code	no_shifts
Logan	Butler	4	7
Ava	Ellis	7	5
Briony	Jones	2	2
Alexis	Butler	3	2
Madison	McConnell	3	2
Joseph	Elrick	3	2
Alexander	Lawson	3	2
David	Jones	2	2
Emily	Best	5	1
Ethan	Hopgood	3	1

- b. Dúos de ingenieros que han trabajado juntos. [Los dos nombres y apellidos y las veces que han trabajado juntos]

```

SELECT
s1.First_name AS E1N,
s1.Last_name AS E1A,
s2.First_name AS E2N,
s2.Last_name AS E2A,
COUNT(*) AS TIMES
FROM Shift sh
JOIN Staff s1 ON sh.Engineer1 = s1.Staff_code
JOIN Staff s2 ON sh.Engineer2 = s2.Staff_code
GROUP BY s1.First_name, s1.Last_name, s2.First_name, s2.Last_name

```

## Result:

E1N	E1A	E2N	E2A	TIMES
Alexander	Lawson	Briony	Jones	1
Alexis	Butler	Briony	Jones	1
Alexis	Butler	David	Jones	1
Ava	Ellis	Joseph	Elrick	1
Ava	Ellis	Michael	Best	1
David	Jones	James	Penrice	1
Ethan	Hopgood	Alexander	Lawson	1
Mia	Ellis	Madison	McConnell	1

- c. Propongan una nueva consulta para esta área: Quienes son los que llaman más y a que nivel se asignan sus Issues?

```

SELECT c.Company_name,
COUNT(i.Call_ref) AS total_calls,
l.Level_code AS "assigned to"
FROM Issue i
JOIN Caller ca ON i.Caller_id = ca.Caller_id
JOIN Customer c ON ca.Company_ref = c.Company_ref
JOIN Staff s ON i.Assigned_to = s.Staff_code
JOIN Level l ON s.Level_code = l.Level_code
GROUP BY c.Company_name, l.Level_code
ORDER BY total_calls DESC;

```

## Result:

Company_name	total_calls	assigned to
Gimmick Inc.	18	3
Hamming Services	13	3
Haunt Services	12	3
High and Co.	11	3
Lady Retail	11	3
Hydrophobic Inc.	11	3
Guttata Shipping	10	3
Immemorial Shipping	10	3

2. De las consultas easy o medium propuestas en SQLZOO, seleccionen y generalicen la que consideren más relevante para este ciclo de desarrollo. Implementarla, si no lo han hecho.

**[medium]:** Show the manager for each shift. Your output should include the shift date and type; also the first and last name of the manager.

```

SELECT Shift.shift_date, Shift.shift_type, Staff.first_name, Staff.last_name
FROM Shift JOIN Staff ON Shift.Manager = Staff.Staff_code
ORDER BY Shift.shift_date;

```

**[medium]:** For each shift show the number of staff assigned. Beware that some roles may be NULL and that the same person might have been assigned to multiple roles (The roles are 'Manager', 'Operator', 'Engineer1', 'Engineer2').

```

select Shift_date, Shift_type, count(distinct role) cw
from (
    select Shift_date, Shift_type, Manager role
    from Shift
    union all

```

```

select Shift_date, Shift_type, Operator role
from Shift
union all
select Shift_date, Shift_type, Engineer1 role
from Shift
union all
select Shift_date, Shift_type, Engineer2 role
from Shift
) query
group by query.Shift_date, query.Shift_type

```

#### 4. PARTE CUATRO. Extendiendo ciclo uno

- a) Show the manager and number of calls received for each hour of the day on 2017-08-12

```

SELECT Manager, HOUR(call_date) Hr, COUNT(*) cc
FROM Issue
JOIN Shift_type st ON Start_time <= hour(call_date) AND hour(call_date) < End_time
JOIN Shift s ON s.Shift_type=st.Shift_type AND Shift_date=DATE(call_date)
WHERE date(call_date) = "2017-08-12"
GROUP BY Manager, Hr
ORDER BY Hr

```

**Para los problemas b, c, d y e:** no logramos implementar las consultas en SQLZoo, ya que requerían conceptos avanzados de SQL, como el uso de variables de usuario (@nombreVariable := valor) y la aplicación de múltiples filtros. Aunque en teoría podrían haberse implementado, la ausencia de estas variables afecta directamente el comportamiento de las consultas y nos impidió resolverlas de manera correcta.

Ante esta dificultad, investigamos cómo podían resolverse y encontramos la siguiente página: <https://cwong8.github.io/projects/sqlzoo/helpdesk> Gracias a ese recurso comprendimos qué funciones y estructuras eran necesarias, aunque decidimos no copiar y pegar las soluciones tal cual aparecen allí.

## RETROSPECTIVA

1. Tiempo total invertido. Jonatan Palomares: 6.5 horas. Juan Diego Patiño: 2.5 horas.
2. La única parte que quedó incompleta fue la cuatro debido a la dificultad de los ejercicios de SQLZOO como ya se mencionó. Por lo demás todo está completo.
3. El mayor logro fue realizar el modelo extendido ya que nos ayudó a comprender mejor cómo tratar con los atributos en astah.
4. El mayor problema fue respecto a las preguntas difíciles de SQLZOO, requerían de algunas funciones de SQL que no habíamos estudiado aún. Aún así revisamos

algunos repositorios, como ya se dijo, para saber qué es lo que se necesita para poder realizar esas consultas.

5. Dado nuestro trabajo en el laboratorio pasado, esta vez intercambiamos los tipos de ejercicios a resolver, por lo que fortalecimos las habilidades que antes pudieron haber quedado a un lado por cada uno en el laboratorio pasado. Algo que podríamos mejorar es realizar el laboratorio con mayor tiempo de anticipación para aumentar la calidad de este.
6. Referencias

*Cwong8.github.io*. (n.d.). <https://cwong8.github.io/projects/sqlzoo/helpdesk>