

Does ChatGPT Write Better SQL Than Human Intelligence?

Dr. Andrea Kennel
Dani Schnider



Dr. Andrea Kennel



Consultant

Lecturer for Databases

Coach for Project Management

University of Applied Sciences
Northwestern Switzerland

Brugg/Windisch, Switzerland



andrea.kennel@fhnw.ch
andrea@infokennel.ch
www.infokennel.ch

About me



Dani Schnider

- Working for Callista
- Oracle ACE Director
- Member of Symposium 42
- Hobby: Craft Beer Brewing



@dani_schnider

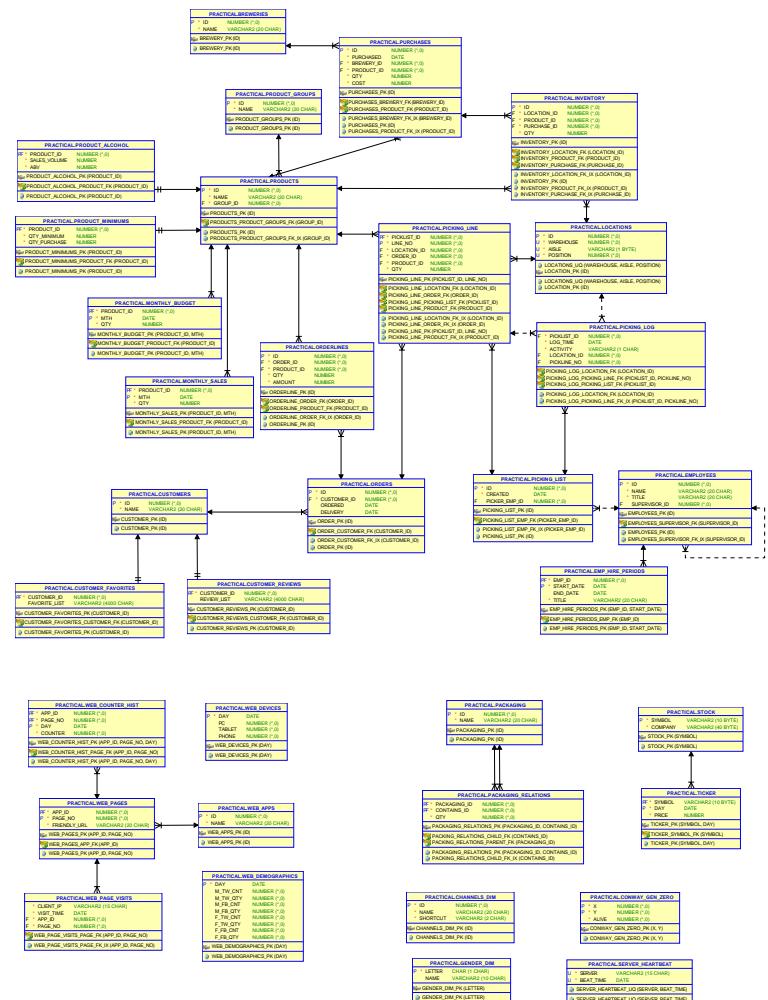
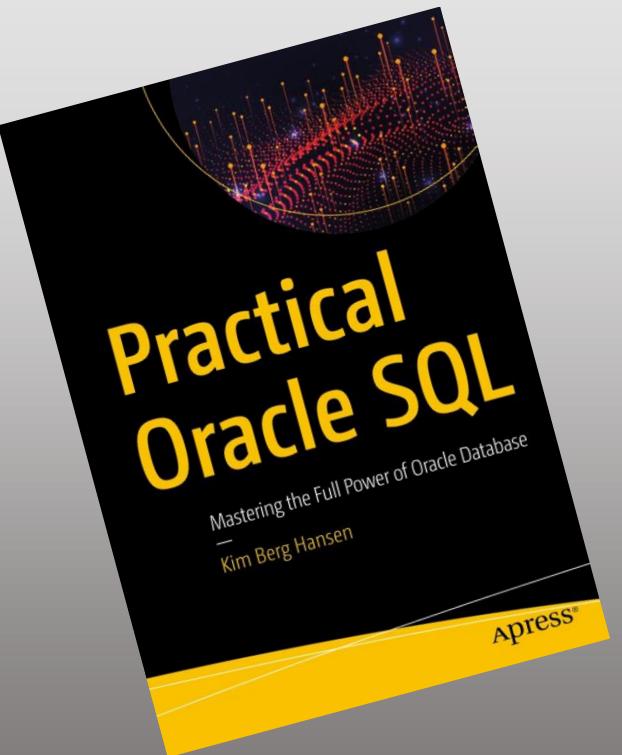


danischnider.wordpress.com



www.linkedin.com/in/danischnider/

Demo Example Schema



<https://github.com/kibeha/practical-oracle-sql/>



ChatGPT ▾

Where is the Baltazar Brewery located?



Baltazar Brewery is located in Mexico City, Mexico. It is a craft brewery known for producing artisanal beers. The brewery focuses on high-quality ingredients and creative brewing techniques. The brewery is part of Mexico City's growing craft beer scene, which has gained international attention.

Hallucination

Would you like to know more about their specific beers or visiting the brewery?

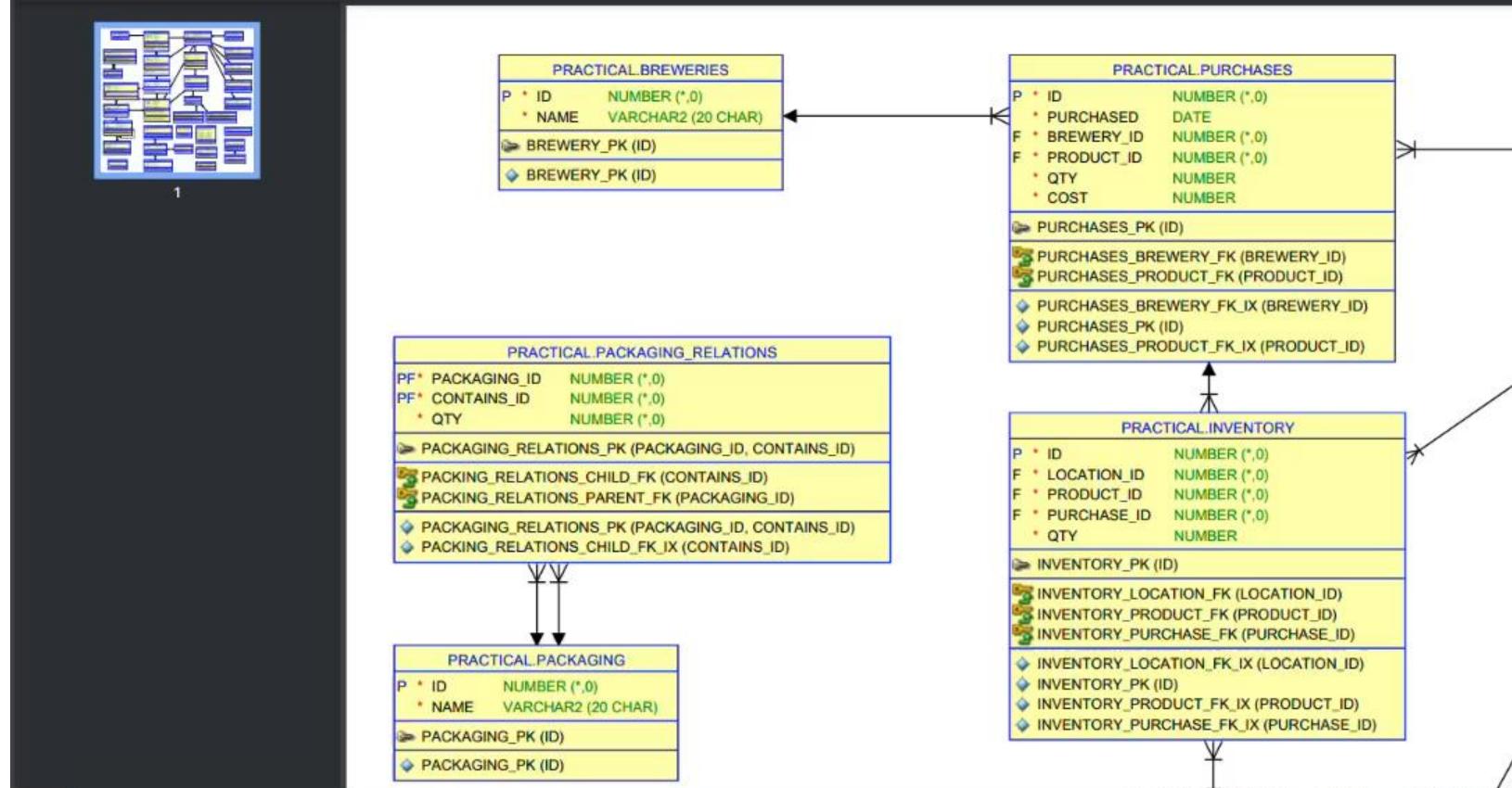
What beers does the Baltazar Brewery produce?



Searched 4 sites ▾

Baltazar Brewery is a craft brewery located in Mexico City, known for its innovative beers and commitment to quality. Among the brews produced by Baltazar Brewery are **Baltazar Blonde**, a crisp and refreshing light beer, and **Baltazar Saison**, a more complex beer with fruity and spicy notes that is often made using local ingredients. These beers reflect the brewery's focus on blending traditional techniques with modern craft beer approaches.

Create a list with names and titles
of all employees sorted by title





Oracle SQL Developer : DB_23.4 - CHATGPT_DEMO

Connections

+ DB_23.4 - CHATGPT_DEMO

- Tables (Filtered)
 - BREWERIES
 - CHANNELS_DIM
 - CONWAY_GEN_ZERO
 - CUSTOMER_FAVORITES
 - CUSTOMER_REVIEWS
 - CUSTOMERS
 - EMP_HIRE_PERIODS
 - EMPLOYEES
 - GENDER_DIM
 - INVENTORY
 - LOCATIONS
 - MONTHLY_BUDGET
 - MONTHLY_SALES
 - ORDERLINES
 - ORDERS
 - PACKAGING
 - PACKAGING_RELATIONS
 - PICKING_LINE
 - PICKING_LIST
 - PICKING_LOG
 - PRODUCT_ALCOHOL
 - PRODUCT_GROUPS
 - PRODUCT_MINIMUMS
 - PRODUCTS
 - PURCHASES
 - SERVER_HEARTBEAT
 - STOCK
 - TICKER
 - WEB_APPS
 - WEB_COUNTER_HIST
 - WEB_DEMOGRAPHICS
 - WEB_DEVICES
 - WEB_PAGE_VISITS
 - WEB_PAGES
- Views
- Indexes

Reports

DB_23.4 - CHATGPT_DEMO

Worksheet Query Builder

```
1 -- List of names and titles of all employees, ordered by title
2 -----
3 -----
4 -----
5 select name, title from employees order by title;
6
```

Query Result

All Rows Fetched: 14 in 0,017 seconds

NAME	TITLE
1 Laura Jensen	Bulk Salesman
2 Kurt Zollman	Forklift Operator
3 Evelyn Smith	Forklift Operator
4 Zoe Thorston	IT Developer
5 Mogens Juel	IT Manager
6 Dan Hoeffler	IT Supporter
7 Susanne Hoff	Janitor
8 Harold King	Managing Director
9 Ursula Mwbesi	Operations Chief
10 Axel de Proef	Product Director
11 Maria Juarez	Purchaser
12 Simon Chang	Retail Salesman
13 Jim Kronzki	Sales Manager
14 Lim Tok Lo	Warehouse Manager

Line 5 Column 50 | Insert | Modified | Unix/Mac: LF

How many employees were employed on
January 1, 2012 and who had which title?



Oracle SQL Developer : DB_23.4 - CHATGPT_DEMO

Connections

DB_23.4 - CHATGPT_DEMO

- Tables (Filtered)
 - BREWERIES
 - CHANNELS_DIM
 - CONWAY_GEN_ZERO
 - CUSTOMER_FAVORITES
 - CUSTOMER_REVIEWS
 - CUSTOMERS
 - EMP_HIRE_PERIODS
 - EMPLOYEES
 - GENDER_DIM
 - INVENTORY
 - LOCATIONS
 - MONTHLY_BUDGET
 - MONTHLY_SALES
 - ORDERLINES
 - ORDERS
 - PACKAGING
 - PACKAGING_RELATIONS
 - PICKING_LINE
 - PICKING_LIST
 - PICKING_LOG
 - PRODUCT_ALCOHOL
 - PRODUCT_GROUPS
 - PRODUCT_MINIMUMS
 - PRODUCTS
 - PURCHASES
 - SERVER_HEARTBEAT
 - STOCK
 - TICKER
 - WEB_APPS
 - WEB_COUNTER_HIST
 - WEB_DEMOGRAPHICS
 - WEB_DEVICES
 - WEB_PAGE_VISITS
 - WEB_PAGES
- Views
- Indexes

Reports

Worksheet Query Builder

```
1 -- How many employees were employed on 1.1.2012, and who had which title?  
2  
3  
4  
5 select e.name, p.title, p.start_date, p.end_date|  
6 from employees e  
7 join emp_hire_periods p on (p.emp_id = e.id)  
8 where date'2012-01-01' between p.start_date and nvl(end_date, sysdate)  
9 order by e.id, p.start_date;  
10
```

Query Result

All Rows Fetched: 3 in 0,015 seconds

	NAME	TITLE	START_DATE	END_DATE
1	Harold King	Product Director	01.07.2010	01.04.2012
2	Mogens Juel	IT Technician	01.07.2010	01.01.2014
3	Axel de Proef	Sales Manager	01.07.2010	01.07.2013

Line 5 Column 49 | Insert | Modified | Unix/Mac: LF



How many employees were employed on January 1, 2012 and who had which title?



To find out how many employees were employed on **January 1, 2012**, along with their titles, we need to query both the `employees` and `emp_hire_periods` tables. Specifically, we will look for employees whose employment periods (determined by `start_date` and `end_date` in `emp_hire_periods`) include January 1, 2012.

Here is the SQL query to achieve that:

sql

Copy code

```
SELECT e.name, e.title
FROM employees e
JOIN emp_hire_periods hp
ON e.id = hp.emp_id
WHERE hp.start_date <= TO_DATE('2012-01-01', 'YYYY-MM-DD')
AND (hp.end_date IS NULL OR hp.end_date >= TO_DATE('2012-01-01', 'YYYY-MM-DD'));
```

Explanation:

- `SELECT e.name, e.title` : This retrieves the `name` and `title` of the employees.



Oracle SQL Developer : ChatGPT_demo

Connections Welcome Page ChatGPT_demo

Worksheet Query Builder

```
SELECT e.name, e.title
FROM employees e
JOIN emp_hire_periods hp
ON e.id = hp.emp_id
WHERE hp.start_date <= TO_DATE('2012-01-01', 'YYYY-MM-DD')
AND (hp.end_date IS NULL OR hp.end_date >= TO_DATE('2012-01-01', 'YYYY-MM-DD'));
```

Query Result

All Rows Fetched: 3 in 0.079 seconds

NAME	TITLE
1 Harold King	Managing Director
2 Mogens Juel	IT Manager
3 Axel de Proef	Product Director

Reports Compiler - Log

Messages Logging page Statements Compiler

Click on an identifier with the Command key down to perform "Go to Declaration"

| Line 7 Column 1 | Insert | Modified | Unix/Mac: LF

Are there any employees
who are no longer employed?



Are there any employees who are no longer employed?



To find employees who are no longer employed, we can query the `emp_hire_periods` table and check for records where the `end_date` is not null. These records represent employees whose employment has ended.

Here's the SQL query to retrieve the names and titles of employees who are no longer employed:

sql

Copy code

```
SELECT e.name, e.title
FROM employees e
JOIN emp_hire_periods hp
ON e.id = hp.emp_id
WHERE hp.end_date IS NOT NULL;
```

Explanation:

- `SELECT e.name, e.title`: Retrieves the `name` and `title` of the employees.
- `JOIN emp_hire_periods hp ON e.id = hp.emp_id`: Joins the `employees` table with the `emp_hire_periods` table based on the employee ID.
- `WHERE hp.end_date IS NOT NULL`: Filters the records where the `end_date` is not null, meaning the employee is no longer employed.



Oracle SQL Developer : ChatGPT_demo

Connections Welcome Page ChatGPT_demo

ChatGPT_demo

Worksheet Query Builder

```
SELECT e.name, e.title
FROM employees e
JOIN emp_hire_periods hp
  ON e.id = hp.emp_id
WHERE hp.end_date IS NOT NULL;
```

Query Result SQL | All Rows Fetched: 8 in 0.083 seconds

NAME	TITLE
1 Harold King	Managing Director
2 Mogens Juel	IT Manager
3 Mogens Juel	IT Manager
4 Mogens Juel	IT Manager
5 Axel de Proef	Product Director
6 Lim Tok Lo	Warehouse Manager
7 Ursula Mwbesi	Operations Chief
8 Ursula Mwbesi	Operations Chief

Reports Compiler - Log
Messages Logging Page Statements Compiler

Click on an identifier with the Command key down to perform "Go to Declaration"

| Line 6 Column 1 | Insert | Modified | Unix/Mac: LF



ChatGPT ▾

The result provides different employees several times. Can you please correct this and also provide the end_date?



To ensure that each employee appears only once in the result and to include their `end_date`, you can use the **DISTINCT** clause or aggregate the results to remove duplicates. Additionally, we'll include the `end_date` in the result to show when each employee's employment ended.

Here's the corrected query:

sql

Copy code

```
SELECT e.name, e.title, MAX(hp.end_date) AS end_date
FROM employees e
JOIN emp_hire_periods hp
ON e.id = hp.emp_id
WHERE hp.end_date IS NOT NULL
GROUP BY e.name, e.title;
```



Oracle SQL Developer : ChatGPT_demo

Connections

Welcome Page ChatGPT_demo

Worksheet Query Builder

```
SELECT e.name, e.title, MAX(hp.end_date) AS end_date
FROM employees e
JOIN emp_hire_periods hp
  ON e.id = hp.emp_id
WHERE hp.end_date IS NOT NULL
GROUP BY e.name, e.title;
```

Query Result

All Rows Fetched: 5 in 0.075 seconds

NAME	TITLE	END_DATE
1 Harold King	Managing Director	01-APR-12
2 Mogens Juel	IT Manager	01-JUN-16
3 Axel de Proef	Product Director	01-JUL-13
4 Lim Tok Lo	Warehouse Manager	01-FEB-16
5 Ursula Mwbesi	Operations Chief	01-MAR-17

Compiler - Log

Messages Logging Page Statements Compiler

Click on an identifier with the Command key down to perform "Go to Declaration"

| Line 7 Column 1 | Insert | Modified | Unix/Mac: LF



Oracle SQL Developer : DB_23.4 - CHATGPT_DEMO

Connections

DB_23.4 - CHATGPT_DEMO

- Tables (Filtered)
 - BREWERIES
 - CHANNELS_DIM
 - CONWAY_GEN_ZERO
 - CUSTOMER_FAVORITES
 - CUSTOMER_REVIEWS
 - CUSTOMERS
 - EMP_HIRE_PERIODS
 - EMPLOYEES
 - GENDER_DIM
 - INVENTORY
 - LOCATIONS
 - MONTHLY_BUDGET
 - MONTHLY_SALES
 - ORDERLINES
 - ORDERS
 - PACKAGING
 - PACKAGING_RELATIONS
 - PICKING_LINE
 - PICKING_LIST
 - PICKING_LOG
 - PRODUCT_ALCOHOL
 - PRODUCT_GROUPS
 - PRODUCT_MINIMUMS
 - PRODUCTS
 - PURCHASES
 - SERVER_HEARTBEAT
 - STOCK
 - TICKER
 - WEB_APPS
 - WEB_COUNTER_HIST
 - WEB_DEMOGRAPHICS
 - WEB_DEVICES
 - WEB_PAGE_VISITS
 - WEB_PAGES
- Views
- Indexes

Worksheet Query Builder

```
1 -- Are there employees who are no longer employed?  
2  
3  
4  
5 select *  
6 from employees e  
7 join emp_hire_periods p on (p.emp_id = e.id)  
8 order by e.id, p.start_date;  
9  
10 select * from employees|  
11 where id not in (select emp_id from emp_hire_periods where end_date is null);  
12
```

Query Result

All Rows Fetched: 8 in 0,022 seconds

ID	NAME	TITLE	SUPERVISOR_ID
1	Maria Juarez	Purchaser	144
2	Kurt Zollman	Forklift Operator	146
3	Laura Jensen	Bulk Salesman	151
4	Jim Kronzki	Sales Manager	144
5	Evelyn Smith	Forklift Operator	146
6	Dan Hoeffler	IT Supporter	143
7	Simon Chang	Retail Salesman	151
8	Susanne Hoff	Janitor	146

Line 10 Column 24 | Insert | Modified | Unix/Mac: LF

Which customer has the highest revenue?

Oracle SQL Developer : DB_23.4 - CHATGPT_DEMO

Connections

DB_23.4 - CHATGPT_DEMO

- Tables (Filtered)
 - BREWERIES
 - CHANNELS_DIM
 - CONWAY_GEN_ZERO
 - CUSTOMER_FAVORITES
 - CUSTOMER_REVIEWS
 - CUSTOMERS
 - EMP_HIRE_PERIODS
 - EMPLOYEES
 - GENDER_DIM
 - INVENTORY
 - LOCATIONS
 - MONTHLY_BUDGET
 - MONTHLY_SALES
 - ORDERLINES
 - ORDERS
 - PACKAGING
 - PACKAGING_RELATIONS
 - PICKING_LINE
 - PICKING_LIST
 - PICKING_LOG
 - PRODUCT_ALCOHOL
 - PRODUCT_GROUPS
 - PRODUCT_MINIMUMS
 - PRODUCTS
 - PURCHASES
 - SERVER_HEARTBEAT
 - STOCK
 - TICKER
 - WEB_APPS
 - WEB_COUNTER_HIST
 - WEB_DEMOGRAPHICS
 - WEB_DEVICES
 - WEB_PAGE_VISITS
 - WEB_PAGES
- Views
- Indexes

Worksheet Query Builder

```
12 with cte_ranking as (
13     select c.name, sum(ol.amount)
14         , rank() over (order by sum(ol.amount) desc) rnk
15     from customers c
16     join orders o on (o.customer_id = c.id)
17     join orderlines ol on (ol.order_id = o.id)
18     group by 1
19 )
20 select *
21 from cte_ranking
22 where rnk = 1;
```

Query Result

All Rows Fetched: 1 in 0,026 seconds

NAME	SUM(OLAMOUNT)	RNK
The White Hart	8755	1

Line 20 Column 9 | Insert | Modified | Unix/Mac: LF





Information for ChatGPT

Our database has even more tables:

```
create table customers (
    id integer constraint customer_pk primary key
    , name varchar2(20 char) not null
);
```

```
create table orders (
    id      integer constraint order_pk primary key
    , customer_id not null constraint order_customer_fk
                    references customers
    , ordered   date
    , delivery  date
);
```

```
create index order_customer_fk_ix on orders
(customer_id);
```

```
create table orderlines (
    id      integer constraint orderline_pk primary key
    , order_id  not null constraint orderline_order_fk
                    references orders
    , product_id not null constraint orderline_product_fk
                    references products
    , qty      number not null
    , amount    number not null
);
```

```
create index orderline_order_fk_ix on orderlines
(order_id);
```

```
create table products (
    id      integer constraint products_pk primary key
    , name    varchar2(20 char) not null
    , group_id not null constraint
products_product_groups_fk
                    references product_groups
);
```



Connections Welcome Page ChatGPT_demo

Worksheet Query Builder

```
SELECT customer_name, total_revenue
FROM (
    SELECT c.name AS customer_name,
           SUM(ol.amount) AS total_revenue,
           RANK() OVER (ORDER BY SUM(ol.amount) DESC) AS revenue_rank
    FROM customers c
    JOIN orders o ON c.id = o.customer_id
    JOIN orderlines ol ON o.id = ol.order_id
    GROUP BY c.id, c.name
) ranked_revenue
WHERE revenue_rank = 1;
```

Query Result SQL | All Rows Fetched: 1 in 0.083 seconds

CUSTOMER_NAME	TOTAL_REVENUE
The White Hart	8755

Reports Compiler - Log
Messages Logging Page Statements Compiler

Click on an identifier with the Command key down to perform "Go to Declaration"

| Line 12 Column 1 | Insert | Modified| Unix/Mac: LF

Which beers are ordered by all customers?



Oracle SQL Developer : ChatGPT_demo

Connections Welcome Page ChatGPT_demo

Worksheet Query Builder

```
SELECT p.name AS beer_name
FROM products p
JOIN orderlines ol ON p.id = ol.product_id
JOIN orders o ON ol.order_id = o.id
GROUP BY p.id, p.name
HAVING COUNT(DISTINCT o.customer_id) = (
    SELECT COUNT(DISTINCT o2.customer_id)
    FROM orders o2
    JOIN orderlines ol2 ON o2.id = ol2.order_id
);
```

Query Result SQL | All Rows Fetched: 2 in 0.071 seconds

BEER_NAME
1 Hoppy Crude Oil
2 Der Helle Kumpel

Reports Compiler - Log Messages Logging Page Statements Compiler

Click on an identifier with the Command key down to perform "Go to Declaration" Line 11 Column 1 Insert Modified Unix/Mac: LF



Connections

+ DB 23.4 - CHATGPT_DEMO

- Tables (Filtered)
 - BREWERIES
 - CHANNELS_DIM
 - CONWAY_GEN_ZERO
 - CUSTOMER_FAVORITES
 - CUSTOMER_REVIEWS
 - CUSTOMERS
 - EMP_HIRE_PERIODS
 - EMPLOYEES
 - GENDER_DIM
 - INVENTORY
 - LOCATIONS
 - MONTHLY_BUDGET
 - MONTHLY_SALES
 - ORDERLINES
 - ORDERS
 - PACKAGING
 - PACKAGING_RELATIONS
 - PICKING_LINE
 - PICKING_LIST
 - PICKING_LOG
 - PRODUCT_ALCOHOL
 - PRODUCT_GROUPS
 - PRODUCT_MINIMUMS
 - PRODUCTS
 - PURCHASES
 - SERVER_HEARTBEAT
 - STOCK
 - TICKER
 - WEB_APPS
 - WEB_COUNTER_HIST
 - WEB_DEMOGRAPHICS
 - WEB_DEVICES
 - WEB_PAGE_VISITS
 - WEB_PAGES
- Views
- Indexes

Click on an identifier with the Command key down to perform "Go to Declaration"

Reports

Worksheet | Query Builder

```
1 -- Which beers are ordered by all customers?
2
3
4
5 with cte_customers_per_beer as (
6     select distinct p.name as beer_name|
7         , count(distinct c.id) over (partition by p.id) as customers_per_beer
8         , count(distinct c.id) over () as customers_total
9     from customers c
10    join orders o on (o.customer_id = c.id)
11    join orderlines ol on (ol.order_id = o.id)
12    join products p on (p.id = ol.product_id)
13 )
14 select * from cte_customers_per_beer
15 where customers_per_beer = customers_total;
16
```

Query Result

SQL | All Rows Fetched: 2 in 0,023 seconds

BEER_NAME	CUSTOMERS_PER_BEER	CUSTOMERS_TOTAL
Hoppy Crude Oil	3	3
Der Helle Kumpel	3	3

What is the average purchase
price per year and beer?



Oracle SQL Developer : DB_23.4 - CHATGPT_DEMO

Connections DB_23.4 - CHATGPT_DEMO

Tables (Filtered)

- BREWERIES
- CHANNELS_DIM
- CONWAY_GEN_ZERO
- CUSTOMER_FAVORITES
- CUSTOMER_REVIEWS
- CUSTOMERS
- EMP_HIRE_PERIODS
- EMPLOYEES
- GENDER_DIM
- INVENTORY
- LOCATIONS
- MONTHLY_BUDGET
- MONTHLY_SALES
- ORDERLINES
- ORDERS
- PACKAGING
- PACKAGING_RELATIONS
- PICKING_LINE
- PICKING_LIST
- PICKING_LOG
- PRODUCT_ALCOHOL
- PRODUCT_GROUPS
- PRODUCT_MINIMUMS
- PRODUCTS
- PURCHASES
- SERVER_HEARTBEAT
- STOCK
- TICKER
- WEB_APPS
- WEB_COUNTER_HIST
- WEB_DEMOGRAPHICS
- WEB_DEVICES
- WEB_PAGE_VISITS
- WEB_PAGES

Views Indexes

Reports

Worksheet Query Builder

```
1 -- What is the average purchase price per year and beer?
2 -----
3 -----
4 -----
5 select * from (
6   select pr.name beer_name
7     , to_char(purchased, 'YYYY') year
8     , round(sum(pu.cost)/sum(pu.qty), 2) avg_purchase_price
9   from purchases pu
10  join products pr on (pr.id = pu.product_id)
11  group by 1, 2
12 )
13 pivot (sum(avg_purchase_price) for year
14 in (2016 as year_2016
15 ,2017 as year_2017
16 ,2018 as year_2018)
17 );
```

Query Result

All Rows Fetched: 10 in 0,02 seconds

BEER_NAME	YEAR_2016	YEAR_2017	YEAR_2018
1 Coalminers Sweat	7.47	5.87	8
2 Reindeer Fuel	6.6	7.73	7.89
3 Hoppy Crude Oil	7.2	5.6	5.2
4 Monks and Nuns	7.52	7.73	6.72
5 Hercule Trippel	6.93	6.8	7.2
6 Der Helle Kumpel	6.56	7.2	7
7 Hazy Pink Cloud	4.8	8	7.2
8 Summer in India	7.4	6.4	6.93
9 Ghost of Hops	7.33	7.68	7.2
10 Pale Rider Rides	7.2	5.6	7.84

Click on an identifier with the Command key down to perform "Go to Declaration"



Oracle SQL Developer : ChatGPT_demo

Welcome Page ChatGPT_demo

Worksheet Query Builder

```
SELECT
    EXTRACT(YEAR FROM o.ordered) AS purchase_year,
    p.name AS beer_name,
    AVG(ol.amount / ol.qty) AS average_price_per_beer
FROM
    orderlines ol
JOIN
    orders o ON ol.order_id = o.id
JOIN
    products p ON ol.product_id = p.id
GROUP BY
    purchase_year, p.name
ORDER BY
    purchase_year, average_price_per_beer
```

Query Results

purchase_year	beer_name	AVERAGE_PRICE_PER_BEER
2019	2019 Kumpel	16.31473214285714285714285714285714
2019	2019 Pink Cloud	20
2019	2019 Hercule Trippel	20.5208334
2019	2019 Hoppy Crude Oil	22.21506734006734006734006734006734006733
2019	2019 Monks and Nuns	18.75
2019	2019 Pale Rider Rides	9.6

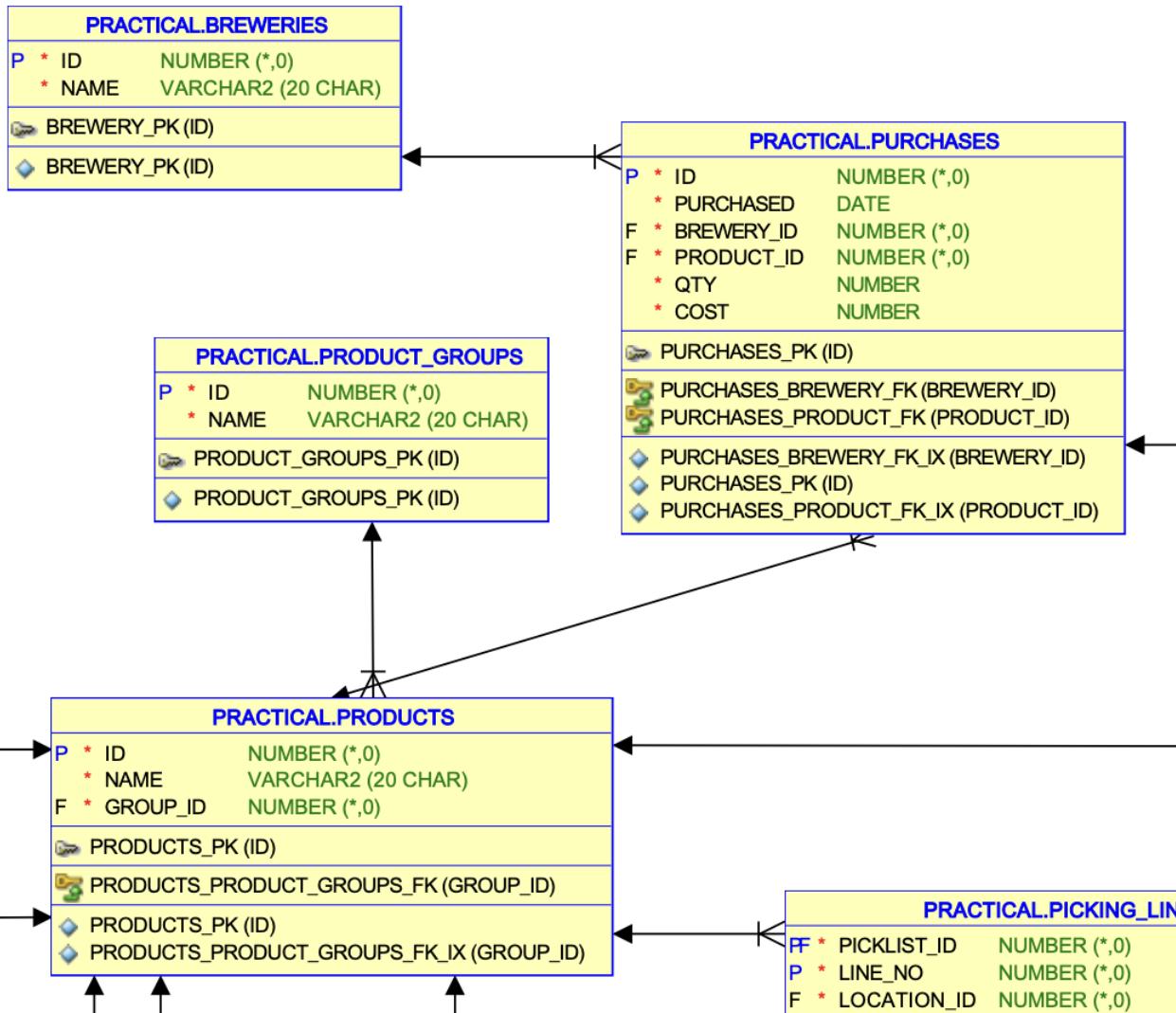
Compiler - Log

Messages Logging Page Statements Compiler

Click on an identifier with the Command key down to perform "Go to Declaration"

purchases unknown

Line 15 Column 1 | Insert | Modified | Unix/Mac: LF





Connections Welcome Page ChatGPT_demo

Worksheet Query Builder

```
SELECT
    EXTRACT(YEAR FROM p.purchased) AS purchase_year,
    pr.name AS beer_name,
    AVG(p.cost / p.qty) AS average_price_per_beer
FROM
    purchases p
JOIN
    products pr ON p.product_id = pr.id
JOIN
    breweries b ON p.brewery_id = b.id
GROUP BY
    purchase_year, pr.id, pr.name
ORDER BY
    purchase_year, pr.name;
```

Query Result All Rows Fetched: 30 in 0.081 seconds

PURCHASE_YEAR	BEER_NAME	AVERAGE_PRICE_PER_BEER
1	2016 Coalminers Sweat	7.46897949383798762666738757313045012478
2	2016 Der Helle Kumpel	6.56148809777651350367044503560376950631
3	2016 Ghost of Hops	7.3343573974113468679543538706964817648
4	2016 Hazy Pink Cloud	4.80151653070643942854896201752185525005
5	2016 Hercule Trippel	6.93557785421323289300668931766653874365
6	2016 Hoppy Crude Oil	7.19783773158386161482136714334856749718
7	2016 Monks and Nuns	7.52191518209756098335748319238106492222

Reports Compiler - Log

Messages Logging Page Statements Compiler

Click on an identifier with the Command key down to perform "Go to Declaration"

| Line 15 Column 1 | Insert | Modified | Unix/Mac: LF



ChatGPT ▾

The formula to calculate the average is wrong and can you round the average to 2 digits?



Thank you for your patience! I now see the issue with the average calculation. The formula I provided earlier divides `cost` by `qty`, but the correct way to calculate the **average price per year and per beer** is to first sum the total costs for each beer and then divide it by the total quantities ordered for that beer during the year.

Here's the corrected SQL query with the proper average calculation, along with rounding the result to two decimal places:

Correct SQL Query:

```
sql Copy code
SELECT
    EXTRACT(YEAR FROM p.purchased) AS purchase_year,
    pr.name AS beer_name,
    ROUND(SUM(p.cost) / SUM(p.qty), 2) AS average_price_per_beer
FROM
    purchases p
JOIN
    products pr ON p.product_id = pr.id
JOIN
    breweries b ON p.brewery_id = b.id
```



Connections Welcome Page ChatGPT_demo

Worksheet Query Builder

```
SELECT
    EXTRACT(YEAR FROM p.purchased) AS purchase_year,
    pr.name AS beer_name,
    ROUND(SUM(p.cost) / SUM(p.qty), 2) AS average_price_per_beer
FROM
    purchases p
JOIN
    products pr ON p.product_id = pr.id
JOIN
    breweries b ON p.brewery_id = b.id
GROUP BY
    purchase_year, pr.id, pr.name
ORDER BY
    purchase_year, pr.name;
```

Query Result All Rows Fetched: 30 in 0.069 seconds

PURCHASE_YEAR	BEER_NAME	AVERAGE_PRICE_PER_BEER
1	2016 Coalminers Sweat	7.47
2	2016 Der Helle Kumpel	6.56
3	2016 Ghost of Hops	7.33
4	2016 Hazy Pink Cloud	4.8
5	2016 Hercule Trippel	6.93
6	2016 Hoppy Crude Oil	7.2
7	2016 Monks and Nuns	7.52
8	2016 Pale Rider Rides	7.2

Reports Compiler - Log Messages Logging Page Statements Compiler

Click on an identifier with the Command key down to perform "Go to Declaration"

| Line 15 Column 1 | Insert | Modified | Unix/Mac: LF

How many days have passed since the
last order per customer and beer?



Connections

DB_23.4 - CHATGPT_DEMO

Tables (Filtered)

- BREWERIES
- CHANNELS_DIM
- CONWAY_GEN_ZERO
- CUSTOMER_FAVORITES
- CUSTOMER_REVIEWS
- CUSTOMERS
- EMP_HIRE_PERIODS
- EMPLOYEES
- GENDER_DIM
- INVENTORY
- LOCATIONS
- MONTHLY_BUDGET
- MONTHLY_SALES
- ORDERLINES
- ORDERS
- PACKAGING
- PACKAGING_RELATIONS
- PICKING_LINE
- PICKING_LIST
- PICKING_LOG
- PRODUCT_ALCOHOL
- PRODUCT_GROUPS
- PRODUCT_MINIMUMS
- PRODUCTS
- PURCHASES
- SERVER_HEARTBEAT
- STOCK
- TICKER
- WEB_APPS
- WEB_COUNTER_HIST
- WEB_DEMOGRAPHICS
- WEB_DEVICES
- WEB_PAGE_VISITS
- WEB_PAGES

Views

Indexes

Reports

Worksheet

Query Builder

```
1 -- What is the number of days per customer and beer since the last order?
2 --
3 --
4
5 select c.name customer_name
6   , o.id order_id
7   , o.ordered
8   , ol.qty
9   , p.name product_name
10  , sum(qty) over (partition by o.id) num_bottles
11  , o.ordered - lag(o.ordered)
12    over (partition by c.name, p.name order by o.id) days_between_orders
13 from customers c
14 join orders o on (o.customer_id = c.id)
15 join orderlines ol on (ol.order_id = o.id)
16 join products p on (p.id = ol.product_id)
17 order by c.name, o.ordered;
```

Query Result

All Rows Fetched: 18 in 0,027 seconds

CUSTOMER_NAME	ORDER_ID	ORDERED	QTY	PRODUCT_NAME	NUM_BOTTLES	DAYS_BETWEEN_ORDERS
1 Der Wichtelmann	42217.01.2019	80	Der Helle Kumpel	160		
2 Der Wichtelmann	42217.01.2019	80	Hoppy Crude Oil	160		
3 Der Wichtelmann	42428.01.2019	16	Hazy Pink Cloud	16		
4 Der Wichtelmann	42517.02.2019	40	Monks and Nuns	124		
5 Der Wichtelmann	42517.02.2019	24	Hazy Pink Cloud	124		
6 Der Wichtelmann	42517.02.2019	60	Hercule Trippel	124		
7 Hygge og Humle	42318.01.2019	40	Der Helle Kumpel	100		
8 Hygge og Humle	42318.01.2019	60	Hoppy Crude Oil	100		
9 Hygge og Humle	42626.02.2019	40	Der Helle Kumpel	56		
						20
						39

Click on an identifier with the Command key down to perform "Go to Declaration"



Oracle SQL Developer : ChatGPT_demo

Connections Welcome Page ChatGPT_demo

Worksheet Query Builder

```
SELECT
    c.name AS customer_name,
    p.name AS product_name,
    (o.ordered - LAG(o.ordered) OVER (PARTITION BY c.id, p.id ORDER BY o.ordered))
FROM
    customers c
JOIN
    orders o ON c.id = o.customer_id
JOIN
    orderlines ol ON o.id = ol.order_id
JOIN
    products p ON ol.product_id = p.id
ORDER BY
    customer name, product name, o.ordered;
```

Query Result | All Rows Fetched: 18 in 0.082 seconds

CUSTOMER_NAME	PRODUCT_NAME	DAYS_BETWEEN_ORDERS
1 Der Wichtelmann	Der Helle Kumpel	(null)
2 Der Wichtelmann	Hazy Pink Cloud	(null)
3 Der Wichtelmann	Hazy Pink Cloud	20
4 Der Wichtelmann	Hercule Trippel	(null)
5 Der Wichtelmann	Hoppy Crude Oil	(null)
6 Der Wichtelmann	Monks and Nuns	(null)
7 Hygge og Humle	Der Helle Kumpel	(null)
8 Hvaea oa Humle	Der Helle Kumpel	39

Reports Compiler - Log
Messages Logging Page Statements Compiler

Click on an identifier with the Command key down to perform "Go to Declaration"

| Line 15 Column 1 | Insert | Modified| Unix/Mac: LF

There is also the table `customer_reviews`
How can I write a query that returns all
reviews for the customer in a clean way?



Ø

There is also the table customer_reviews with the following data:

CUSTOMER_ID REVIEW_LIST

50042 4040:A,6600:C,7950:B

51069 4280:B,7790:B

50741 4160:A

51007

How can I write a query that returns all reviews for the customer in a clean way?



Connections Welcome Page ChatGPT_demo

Worksheet Query Builder

```
SELECT
    cr.customer_id,
    SUBSTR(REGEXP_SUBSTR(cr.review_list, '[^,]+', 1, LEVEL), 1, INSTR(REGEXP_SUBSTR(cr.review_list, '[^,]+', 1, LEVEL), INSTR(REGEXP_SUBSTR(c
FROM
    customer_reviews cr
CONNECT BY
    PRIOR cr.customer_id = cr.customer_id
    AND PRIOR SYS_GUID() IS NOT NULL
    AND LEVEL <= REGEXP_COUNT(cr.review_list, ',') + 1
ORDER BY
    cr.customer_id, product_id;
```

Query Result

All Rows Fetched: 7 in 0.074 seconds

	CUSTOMER_ID	PRODUCT_ID	REVIEW_GRADE
1	50042 4040	A	
2	50042 6600	C	
3	50042 7950	B	
4	50741 4160	A	
5	51007 (null)	(null)	
6	51069 4280	B	
7	51069 7790	B	

Reports Compiler - Log
Messages Logging Page Statements Compiler

| Line 13 Column 1 | Insert | Modified | Unix/Mac: LF



Oracle SQL Developer : DB_23.4 - CHATGPT_DEMO

Connections

DB_23.4 - CHATGPT_DEMO

- Tables (Filtered)
 - BREWERIES
 - CHANNELS_DIM
 - CONWAY_GEN_ZERO
 - CUSTOMER_FAVORITES
 - CUSTOMER_REVIEWS
 - CUSTOMERS
 - EMP_HIRE_PERIODS
 - EMPLOYEES
 - GENDER_DIM
 - INVENTORY
 - LOCATIONS
 - MONTHLY_BUDGET
 - MONTHLY_SALES
 - ORDERLINES
 - ORDERS
 - PACKAGING
 - PACKAGING_RELATIONS
 - PICKING_LINE
 - PICKING_LIST
 - PICKING_LOG
 - PRODUCT_ALCOHOL
 - PRODUCT_GROUPS
 - PRODUCT_MINIMUMS
 - PRODUCTS
 - PURCHASES
 - SERVER_HEARTBEAT
 - STOCK
 - TICKER
 - WEB_APPS
 - WEB_COUNTER_HIST
 - WEB_DEMOGRAPHICS
 - WEB_DEVICES
 - WEB_PAGE_VISITS
 - WEB_PAGES
- Views
- Indexes

Reports

DB_23.4 - CHATGPT_DEMO

Worksheet Query Builder

```
1 -- Formatted output of customer reviews
2
3
4
5 select cr.customer_id, rl.product_id, rl.score
6 from customer_reviews cr
7 cross apply json_table (
8   nvl2(cr.review_list, '[{"p":', null)
9   || replace(
10     replace(
11       replace(cr.review_list, ',', '|')
12       , ':', ',r":')
13       , '|', '}},{"p":')
14   || nvl2(cr.review_list, '}]', null), '$[*]')
15   columns ( product_id number path '$.p'
16             , score      varchar2(1) path '$.r')
17 ) rl
```

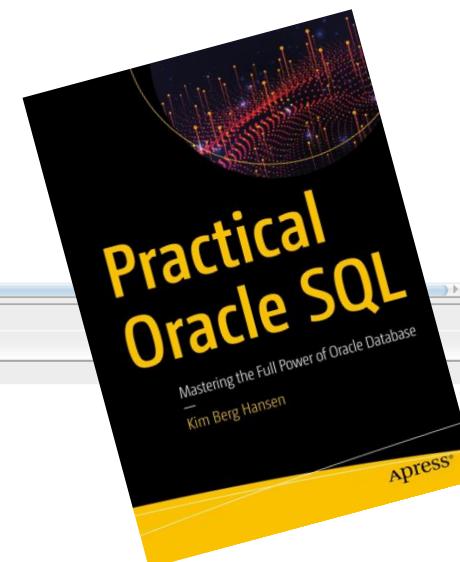
Query Result

All Rows Fetched: 6 in 0,016 seconds

	CUSTOMER_ID	PRODUCT_ID	SCORE
1	50042	4040A	
2	50042	6600C	
3	50042	7950B	
4	51069	4280B	
5	51069	7790B	
6	50741	4160A	

Click on an identifier with the Command key down to perform "Go to Declaration"

Line 7 Column 25 | Insert | Modified | Unix/Mac: LF

The image shows the front cover of the book "Practical Oracle SQL" by Kim Berg Hansen. The cover is black and yellow, featuring a stylized graphic of glowing particles or data points forming a wave-like pattern at the top. The title "Practical Oracle SQL" is written in large, bold, yellow letters. Below the title, it says "Mastering the Full Power of Oracle Database". The author's name, "Kim Berg Hansen", is at the bottom. The Apress logo is in the bottom right corner.



Does ChatGPT learns within the session?



ChatGPT ▾

Where is the Baltazar Brewery located?



Searched 4 sites ▾

Baltazar Brewery, also known as **Pivovara Medvedgrad**, is a prominent Croatian craft brewery based in **Zagreb**, Croatia. Medvedgrad is renowned for its traditional brewing methods and natural lagers, and it has gained popularity since its founding in 1994. Baltazar is one of their many celebrated beers(Untappd)(Croatia Yachting Charter).

What beers does the Baltazar Brewery produce?



Searched 5 sites ▾

The Baltazar Brewery, part of Pivovara Medvedgrad, produces a range of beers with diverse styles. Some of their notable beers include:

- **CugMajstor:** An imperial pilsner with an alcohol content of 6.1%, brewed with noble hops like Aurora and Tettnanger.
- **Pivo Svetog Martina:** A barleywine with a strong malty flavor, boasting 10.3% ABV.



Within the session, ChatGPT learns

ChatGPT ▾



Wo befindet sich die Balthazar Brauerei?



Da deine Tabelle `breweries` nur die `id` und `name` der Brauereien enthält, kannst du keine Standortdaten direkt aus dieser Tabelle abfragen. Wenn du Informationen zum Standort der Balthazar Brauerei benötigst, aber nur diese Tabelle zur Verfügung hast, bedeutet das, dass diese Standortdaten in einer anderen Tabelle oder Datenquelle vorhanden sein müssen.

Wenn keine andere Tabelle vorhanden ist und du nur diese Tabelle hast, dann kannst du die Standortdaten nicht abfragen. Du müsstest diese Informationen entweder in deiner Datenbank ergänzen oder sie aus einer externen Quelle beziehen.

But only in German ;-)

Conclusion



Conclusion



- Simple and short queries
 - Humans are faster
- Semantics and understanding of data
 - Humans are better at analyzing data and adapting queries
 - Humans certainly know better what a beer is
 - ChatGPT must be explained the entire semantics
- Semantics and understanding of the question
 - Both can misunderstand questions
- In general and performance
 - Humans think about performance
 - ChatGPT does what is most frequently found
- ChatGPT provides good solution approaches,
human can judge what is correct



Quelle: <https://byo.com/article/keep-it-simple-its-a-smash/>