

# Schreibt ChatGPT intelligentere SQL-Abfragen als ein Mensch?

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](mailto:andrea.kennel@fhnw.ch)  
[andrea@infokennel.ch](mailto:andrea@infokennel.ch)  
[www.infokennel.ch](http://www.infokennel.ch)

# About me



## Dani Schnider

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



@dani\_schnider

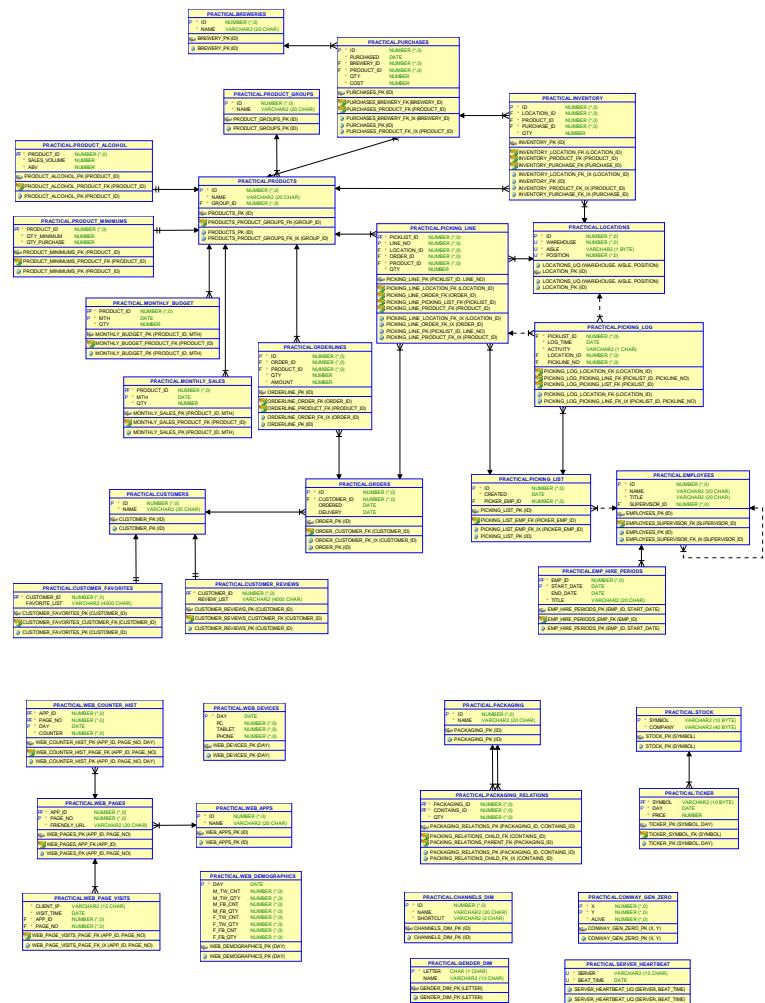
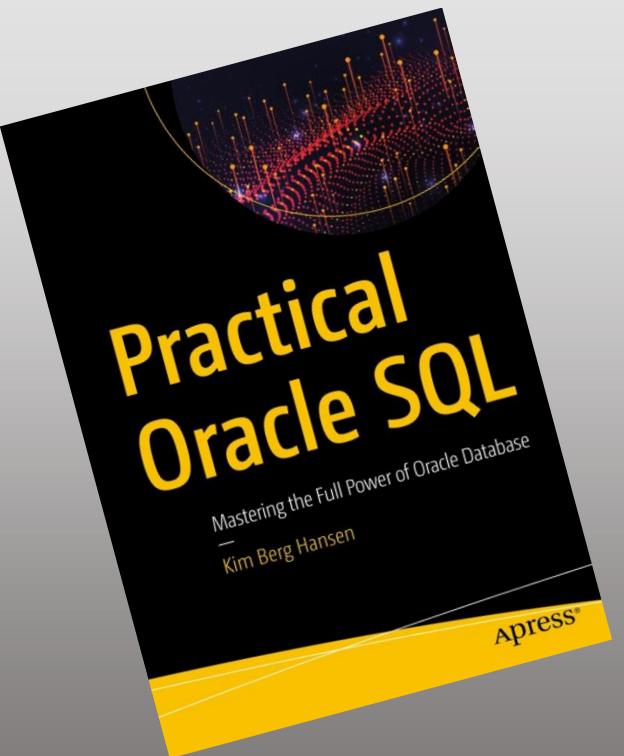


[danischnider.wordpress.com](http://danischnider.wordpress.com)



[www.linkedin.com/in/danischnider/](https://www.linkedin.com/in/danischnider/)

# Demo Example Schema



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



ChatGPT ▾

Wo befindet sich die Balthazar Brauerei?



Die **Balthazar Brauerei** befindet sich in der Stadt **Pirmasens** in Rheinland-Pfalz, Deutschland. Sie ist bekannt für ihr handwerklich gebrautes Bier und liegt in der Nähe des Pfälzerwaldes, einer Region, die für ihre traditionelle Tradition geschätzt wird.

# Halluzination

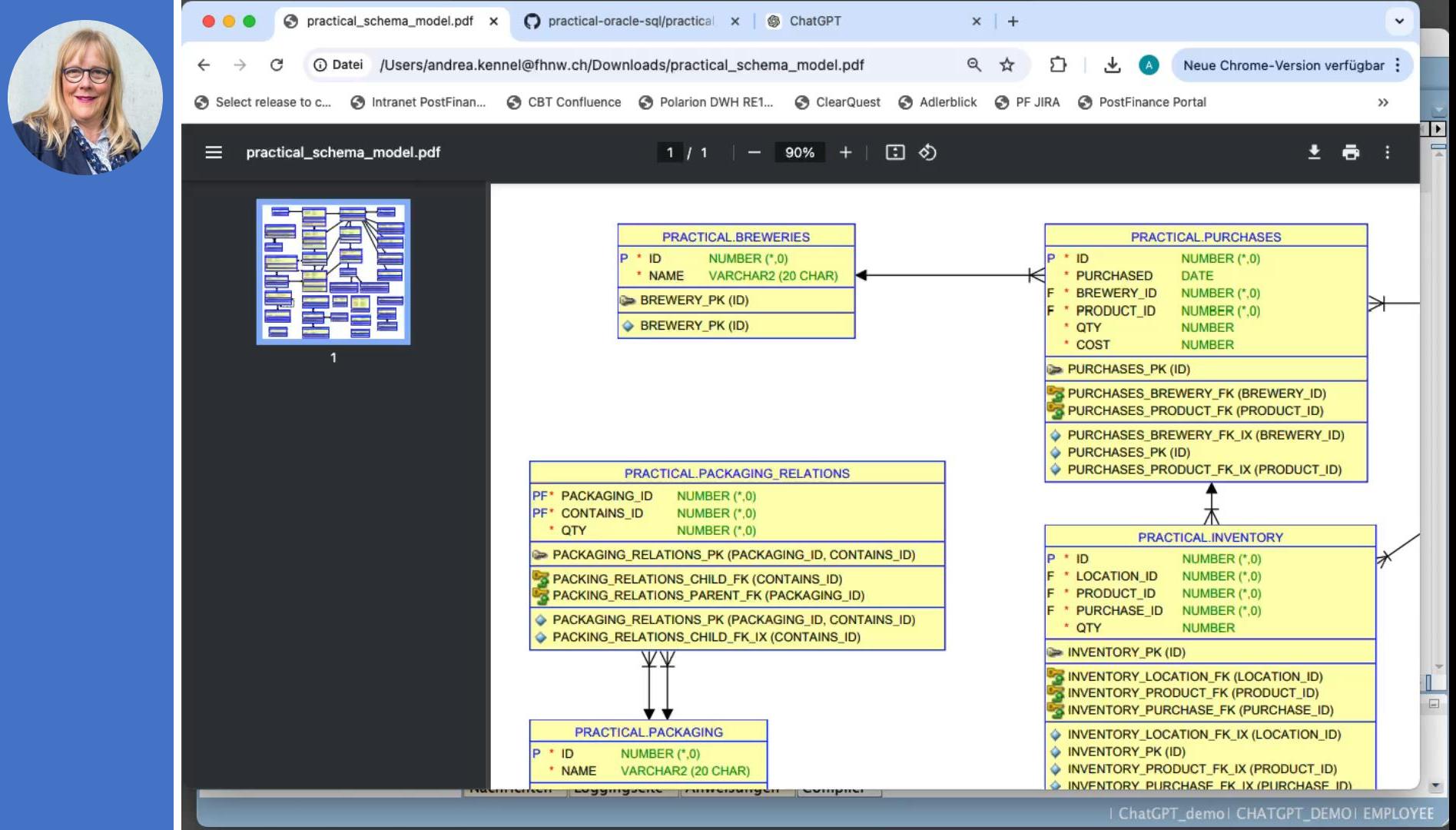


Die **Balthazar Brauerei** in Pirmasens produziert eine Reihe von handwerklich gebrauten Bieren, die sich durch Kreativität und Qualität auszeichnen. Einige der bekannten Biere aus ihrem Sortiment sind:

1. **Balthazar Pale Ale** – Ein hopfenbetontes, fruchtiges Ale, das dem klassischen Pale Ale-Stil folgt.
2. **Balthazar IPA** – Ein intensiveres India Pale Ale mit kräftigen Hopfennoten und einer höheren Bitterkeit.



Erstelle eine Liste mit Namen und  
Titel aller Mitarbeiter sortiert nach Titel





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

Wie viele Mitarbeiter waren am 1.1.2012  
angestellt und wer hatte welchen Titel?



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



ChatGPT ▾

Wie viele Employees waren am 1.1.2012 angestellt und wer hatte welchen Titel?



Um herauszufinden, wie viele Employees am 1. Januar 2012 angestellt waren und welche Titel sie zu diesem Zeitpunkt hatten, musst du die Tabelle `emp_hire_periods` verwenden. Hier speichert die Spalte `start_date`, wann ein Employee seine Anstellung begonnen hat, und `end_date`, wann sie endete (falls sie beendet ist). Du kannst eine Abfrage formulieren, die überprüft, ob der 1. Januar 2012 innerhalb der Beschäftigungszeiträume liegt.

Die entsprechende SQL-Abfrage sieht folgendermaßen aus:

sql

Code kopieren

```
SELECT e.name, ehp.title
FROM employees e
JOIN emp_hire_periods ehp ON e.id = ehp.emp_id
WHERE DATE '2012-01-01' BETWEEN ehp.start_date AND NVL(ehp.end_date, DATE '9999-12-
```



Erläuterung der Abfrage:



Oracle SQL Developer : ChatGPT\_demo

Verbindungen

ChatGPT\_demo

- + ChatGPT\_demo
  - + Tabellen (Gefiltert)
    - + BREWERIES
    - + CHANNELS\_DIM
    - + CONWAY\_GEN\_ZERO
    - + CUSTOMER\_FAVORITES
    - + CUSTOMER\_REVIEWS
    - + CUSTOMERS
    - + EMP\_HIRE\_PERIODS
    - + EMPLOYEES
    - + GENDER\_DIM
    - + INVENTORY
    - + LOCATIONS
    - + MONTHLY\_BUDGET
    - + MONTHLY\_SALES
    - + ORDERINES

Arbeitsblatt Query Builder

```
SELECT e.name, ehp.title
FROM employees e
JOIN emp_hire_periods ehp ON e.id = ehp.emp_id
WHERE DATE '2012-01-01' BETWEEN ehp.start_date AND NVL(ehp.end_date, DATE '9999-12-31');
```

Abfrageergebnis

SQL | Alle Zeilen abgerufen:3 in 0.094 Sekunden

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

Compiler - Log

Nachrichten Loggingseite Anweisungen Compiler

| Zeile 1 Spalte 3 | Einfügen | Geändert! Unix/Mac: LF |

Gibt es Mitarbeitende,  
die nicht mehr angestellt sind?



ChatGPT ▾



Um herauszufinden, ob es Mitarbeitende gibt, die nicht mehr angestellt sind, kannst du nach Einträgen in der Tabelle `emp_hire_periods` suchen, bei denen das Feld `end_date` gesetzt ist, also nicht `NULL` ist. Dies zeigt an, dass das Anstellungsverhältnis beendet wurde.

Die SQL-Abfrage sieht folgendermaßen aus:

sql Code kopieren

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

### Erläuterung der Abfrage:

1. `JOIN employees e ON e.id = ehp.emp_id` : Dies verbindet die Tabellen `employees` und `emp_hire_periods`, um die Namen und Titel der Mitarbeitenden abzurufen.
2. `WHERE ehp.end_date IS NOT NULL` : Diese Bedingung filtert die Datensätze heraus, bei denen ein `end_date` vorhanden ist, was bedeutet, dass der Mitarbeiter nicht mehr angestellt ist.  




Oracle SQL Developer : ChatGPT\_demo

Verbindungen

ChatGPT\_demo

Arbeitsblatt Query Builder

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

Abfrageergebnis

Alle Zeilen abgerufen: 8 in 0.095 Sekunden

	NAME	TITLE	END_DATE
1	Harold King	Product Director	01.04.12
2	Mogens Juel	Sys Admin	01.06.16
3	Mogens Juel	IT Technician	01.01.14
4	Mogens Juel	Code Tester	01.10.15
5	Axel de Proef	Sales Manager	01.07.13
6	Lim Tok Lo	Forklift Operator	01.02.16
7	Ursula Mwbesi	Delivery Manager	01.05.15
8	Ursula Mwbesi	Warehouse Manager	01.03.17

Compiler - Log

Nachrichten Logging Seite Anweisungen Compiler

| Zeile 3 Spalte 10 | Einfügen | Geändert | Unix/Mac: LF



Das Resultat liefert mehrere Mitarbeitende mehrfach mit unterschiedlichen end\_date. Kannst Du das bitte korrigieren?



Das Problem tritt auf, weil in der Tabelle `emp_hire_periods` mehrere Anstellungszeiträume für denselben Mitarbeiter gespeichert sind, was dazu führt, dass derselbe Mitarbeiter mehrfach mit verschiedenen `end_date`-Werten erscheint. Um das zu korrigieren und nur diejenigen Mitarbeitenden zu finden, die aktuell nicht mehr angestellt sind, können wir die letzte Anstellungsperiode eines Mitarbeiters abfragen (also das maximale `end_date`), und dann prüfen, ob diese Periode beendet ist.

Hier ist die korrigierte SQL-Abfrage:

```
sql
SELECT e.name, ehp.title, ehp.end_date
FROM employees e
JOIN emp_hire_periods ehp ON e.id = ehp.emp_id
WHERE ehp.end_date = (
    SELECT MAX(sub_ehp.end_date)
    FROM
        (
```

Code kopieren





Oracle SQL Developer : ChatGPT\_demo

Verbindungen ChatGPT\_demo

Arbeitsblatt Query Builder

```
FROM employees e
JOIN emp_hire_periods ehp ON e.id = ehp.emp_id
WHERE ehp.end_date = (
    SELECT MAX(sub_ehp.end_date)
    FROM emp_hire_periods sub_ehp
    WHERE sub_ehp.emp_id = e.id
)
AND ehp.end_date IS NOT NULL;
```

Abfrageergebnis

Alle Zeilen abgerufen: 5 in 0.098 Sekunden

	NAME	TITLE	END_DATE
1	Harold King	Product Director	01.04.12
2	Mogens Juel	Sys Admin	01.06.16
3	Axel de Proef	Sales Manager	01.07.13
4	Lim Tok Lo	Forklift Operator	01.02.16
5	Ursula Mwbesi	Warehouse Manager	01.03.17

Compiler – Log

Inhalt Loggingseite Anweisungen Compiler

| Zeile 9 Spalte 30 | Einfügen | Geändert! 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

# Welcher Kunde hat den höchsten Umsatz?



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 -- Which customer has the highest turnover?
2
3
4
5 select c.name, sum(ol.amount)
6 from customers c
7 join orders o on (o.customer_id = c.id)
8 join orderlines ol on (ol.order_id = o.id)
9 group by 1 order by 2 desc
10 fetch first 1 rows with ties;
11
```

Query Result

All Rows Fetched: 1 in 0,022 seconds

NAME	SUM(OL.AMOUNT)
The White Hart	8755

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

Line 10 Column 20 | Insert | Modified | Unix/Mac: LF



# Hinweis für ChatGPT

Unsere Datenbank hat noch weitere Tabellen:

```
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
);
```



Oracle SQL Developer : ChatGPT\_demo

Verbindungen

ChatGPT\_demo

Arbeitsblatt Query Builder

```
SELECT name, total_revenue
FROM (
    SELECT c.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.name
)
WHERE revenue_rank = 1;
```

Berichte

Alle Berichte

- Analyse-View – Berichte
- Benutzerdefinierte Berichte
- Data Dictionary-Berichte
- Data Modeler-Berichte
- OLAP-Berichte
- TimesTen-Berichte

Abfrageergebnis

SQL | Alle Zeilen abgerufen:1 in 0.069 Sekunden

NAME	TOTAL_REVENUE
The White Hart	8755

Compiler – Log

Nachrichten Loggingseite Anweisungen Compiler

| Zeile 4 Spalte 70 | Einfügen | Geändert! Unix/Mac: LF

Welche Biere werden  
von allen Kunden bestellt?



Oracle SQL Developer : ChatGPT\_demo

Verbindungen

ChatGPT\_demo

Arbeitsblatt | Query Builder

```
WITH customers_who_ordered_beers AS (
    SELECT DISTINCT o.customer_id
    FROM orders o
    JOIN orderlines ol ON o.id = ol.order_id
),
beers_ordered_by_customers AS (
    SELECT ol.product_id
    FROM orderlines ol
    JOIN orders o ON ol.order_id = o.id
    WHERE o.customer_id IN (SELECT customer_id FROM customers_who_ordered_beers)
    GROUP BY ol.product_id
    HAVING COUNT(DISTINCT o.customer_id) =
        (SELECT COUNT(*) FROM customers_who_ordered_beers)
)
SELECT p.name AS beer_name
FROM products p
WHERE p.id IN (SELECT product_id FROM beers_ordered_by_customers);
```

Skriptausgabe x Abfrageergebnis x

SQL | Alle Zeilen abgerufen: 2 in 0.075 Sekunden

BEER_NAME
1 Hoppy Crude Oil
2 Der Helle Kumpel

Zum Starten von "Gehe zu Deklaration" klicken Sie mit gedrückter Befehlstaste auf einen Identifier

| Zeile 17 Spalte 67 | Einfügen | Geändert! 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

Reports

DB\_23.4 - CHATGPT\_DEMO

Worksheet

```
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

Was ist der durchschnittliche  
Einkaufspreis je Jahr und Bier?



Oracle SQL Developer : DB\_23.4 - CHATGPT\_DEMO

Connections    DB\_23.4 - CHATGPT\_DEMO

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

Verbindungen

ChatGPT\_demo

Arbeitsblatt Query Builder

```
SELECT p.name AS beer_name,
       EXTRACT(YEAR FROM pu.purchased) AS year,
       ROUND(AVG(pu.cost / pu.qty), 2) AS avg_purchase_price
  FROM products p
 JOIN purchases pu ON p.id = pu.product_id
 GROUP BY p.name, EXTRACT(YEAR FROM pu.purchased)
 ORDER BY beer_name, year;
```

Berichte

Alle Berichte

- Analyse-View – Berichte
- Benutzerdefinierte Berichte
- Data Dictionary-Berichte
- Data Modeler-Berichte
- OLAP-Berichte
- TimesTen-Berichte

Skriptausgabe x | Abfrageergebnis x

SQL | Alle Zeilen abgerufen:30 in 0.09 Sekunden

BEER_NAME	YEAR	AVG_PURCHASE_PRICE
1 Coalminers Sweat	2016	7.47
2 Coalminers Sweat	2017	5.87
3 Coalminers Sweat	2018	8
4 Der Helle Kumpel	2016	6.56
5 Der Helle Kumpel	2017	7.2
6 Der Helle Kumpel	2018	7

Compiler - Log

Nachrichten Loggingseite Anweisungen Compiler

| Zeile 3 Spalte 61 | Einfügen | Geändert! Unix/Mac: LF

Wie viele Tage sind seit der letzten  
Bestellung pro Kunde und Bier vergangen?



Connections

DB\_23.4 - CHATGPT\_DEMO

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

Verbindungen

ChatGPT\_demo

Arbeitsblatt Query Builder

```
SELECT c.name AS customer_name,
       p.name AS product_name,
       o.ordered AS current_order_date,
       LAG(o.ordered) OVER (PARTITION BY c.id, p.id ORDER BY o.ordered) AS previous_order_date,
       o.ordered - LAG(o.ordered) OVER (PARTITION BY c.id, p.id ORDER BY o.ordered) AS days_between_orders
  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 c.name, p.name, o.ordered;
```

Skriptausgabe x Abfrageergebnis x

Alle Zeilen abgerufen: 18 in 0.078 Sekunden

CUSTOMER_NAME	PRODUCT_NAME	CURRENT_ORDER_DATE	PREVIOUS_ORDER_DATE	DAYS_BETWEEN_ORDERS
1 Der Wichtelmann	Der Helle Kumpel	17.01.19	(null)	(null)
2 Der Wichtelmann	Hazy Pink Cloud	28.01.19	(null)	(null)
3 Der Wichtelmann	Hazy Pink Cloud	17.02.19	28.01.19	20
4 Der Wichtelmann	Hercule Trippel	17.02.19	(null)	(null)
5 Der Wichtelmann	Hoppy Crude Oil	17.01.19	(null)	(null)
6 Der Wichtelmann	Monks and Nuns	17.02.19	(null)	(null)
7 Hygge og Humle	Der Helle Kumpel	18.01.19	(null)	(null)
8 Hygge og Humle	Der Helle Kumpel	26.02.19	18.01.19	39
9 Hygge og Humle	Hazy Pink Cloud	26.02.19	(null)	(null)
10 Hygge og Humle	Hoppy Crude Oil	18.01.19	(null)	(null)
11 Hygge og Humle	Hoppy Crude Oil	12.03.19	18.01.19	53
12 Hygge og Humle	Pale Rider Rides	12.03.19	(null)	(null)

Compiler - Log

Zum Starten von "Gehe zu Deklaration" klicken Sie mit gedrückter Befehlstaste auf einen Identifier | Zeile 11 Spalte 36 | Einfügen | Geändert! Unix/Mac: LF

Es gibt auch die Tabelle `customer_reviews`  
Wie kann ich eine Abfrage schreiben, die mir  
sauber alle Reviews für die `customer` ausgibt?



∅

Es gibt auch die Tabelle customer\_reviews mit folgenden Daten: ...

CUSTOMER\_ID REVIEW\_LIST

---

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

51069 4280:B,7790:B

50741 4160:A

51007

Wie kann ich eine Abfrage schreiben, die mir sauber alle Reviews für die customer ausgibt?



Oracle SQL Developer : ChatGPT\_demo

Verbindungen ChatGPT\_demo

ChatGPT\_demo

Arbeitsblatt Query Builder

```
WITH SplitReviews AS (
    SELECT customer_id,
        REGEXP_SUBSTR(review_list, '[^,]+', 1, LEVEL) AS review
    FROM customer_reviews
    CONNECT BY REGEXP_SUBSTR(review_list, '[^,]+', 1, LEVEL) IS NOT NULL
    AND PRIOR customer_id = customer_id
    AND PRIOR SYS_GUID() IS NOT NULL
)
SELECT customer_id,
    SUBSTR(review, 1, INSTR(review, ':') - 1) AS product_id,
    SUBSTR(review, INSTR(review, ':') + 1) AS review_grade
FROM SplitReviews
ORDER BY customer_id, product_id;
```

Berichte

Alle Berichte

- Analyse-View – Berichte
- Benutzerdefinierte Berichte
- Data Dictionary-Berichte
- Data Modeler-Berichte
- OLAP-Berichte
- TimesTen-Berichte

Skriptausgabe x Abfrageergebnis x

SQL | Alle Zeilen abgerufen: 7 in 0.059 Sekunden

	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	

Zum Starten von "Gehe zu Deklaration" klicken Sie mit gedrückter Befehlstaste auf einen Identifier | Zeile 13 Spalte 34 | Einfügen | Geändert! Unix/Mac: LF

Oberdorfstrasse 15



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
```

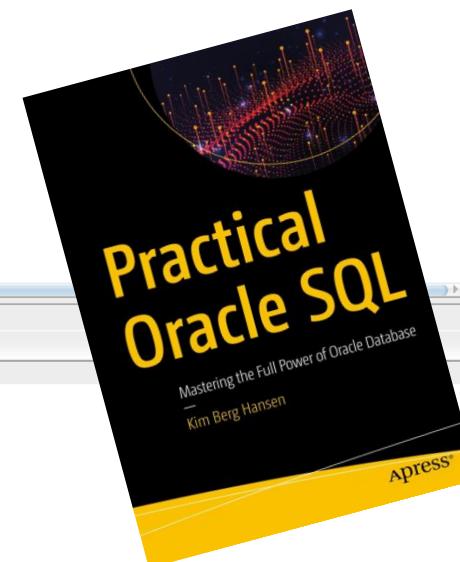
Query Result

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.



# Innerhalb der Session lernt ChatGPT

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.

# Fazit



# Fazit



- Einfache und kurze Abfragen
  - Der Mensch ist schneller
- Semantik und Verständnis der Daten
  - Der Mensch kann Daten besser analysieren und Abfragen anpassen
  - Was ein Bier ist, weiss der Mensch sicher besser
  - ChatGPT muss die ganze Semantik erklärt werden
- Semantik und Verständnis der Frage
  - Beide können Fragen falsch verstehen
- Generell und Performance
  - Mensch überlegt
  - ChatGPT macht so, wie am häufigsten gefunden
- ChatGPT liefert gut Lösungsansätze,  
Mensch kann einschätzen, was korrekt ist

