

Beschreibung der Tabellen und ihrer Beziehungen

Das Ziel unserer Datenbank besteht darin, dass ein Mitarbeiter, welcher seinen Dienst in unserem Krankenhaus gerade zum ersten Mal angetreten hat und noch keinen Überblick der Tagesstruktur besitzt, anhand der Datenbank alle notwendigen Informationen abrufen kann, um den laufenden Betrieb nicht unterbrechen zu müssen.



Tabellen werden in „**Blau**“ dargestellt!

Krankenhaus-Kontaktinfos (Impressum)

1. Über die Krankenhaus-Webseite, die auf **Krankenhaus** zugreift, können potenzielle Patienten die Adresse und weitere Kontaktinfos von dem Krankenhaus erfahren.

Persönliche Daten der Patienten

1. Alle relevanten Daten neuer Patienten (darunter auch die Krankengeschichte) werden in „**Patient**“ und „**VersicherungsStatus**“ registriert.

Aufgaben und Tätigkeiten

1. Für Patienten können Behandlungen oder Laboruntersuchungen geplant werden. Diese sind Arten von **Tätigkeiten**. Alle Voraussetzungen für das Ausüben der jeweiligen Tätigkeit, einschließlich Anzahl und Art von Mitarbeitern, Art und Ausstattung von Räumen, sowie die voraussichtliche Dauer und alle weiteren benötigten Hilfsmittel können dieser und den verbundenen Tabellen entnommen werden.
2. Eine geplante Tätigkeit wird zu einer Aufgabe und wird im **Aufgabenplan** vermerkt. Dort werden die zugehörigen Mitarbeiter, Räume, Patienten und ein Termin (Startzeit) zugeordnet.
3. Aufgaben mit einem Termin, dessen Terminierung in der Zukunft geplant sind.
Aufgaben mit einem Termin in der Vergangenheit sind bereits durchgeführt worden.
4. Wenn der Termin einer Aufgabe in der Vergangenheit liegt und schon mehr Zeit vergangen ist, als für das Durchführen der Aufgabe nötig ist, kann es trotzdem sein, dass die Aufgabe wegen Verzögerungen noch nicht beendet worden ist. Aus diesem Grund gibt es die Flagge „**erledigt**“, die von den Mitarbeitern gesetzt werden muss, nachdem sie eine Aufgabe abgeschlossen haben. Erst danach gelten die involvierten Mitarbeiter, Räume und Patienten als **frei** und können weiter verplant werden.

```
1 • SELECT * FROM krankenhaus.taetigkeit;
```

idTaetigkeit	TaetigkeitenBezeichnung	Dauer	Preis	TaetigkeitenKategorie_idTaetigkeitenKategorie	brauchtPatienten	reserviertPatienten
1	HerzOperation	04:00:00	20000	1	1	NULL
2	ZimmerReinigung	00:15:00	NULL	4	0	NULL
3	PatientenPflege	08:00:00	NULL	1	0	NULL
4	erholung	08:00:00	200	1	1	NULL
5	GluehbirnenWechseln	00:01:00	NULL	3	0	NULL
6	MittagsPause	01:00:00	NULL	0	0	NULL
7	Buchhaltung	08:00:00	NULL	5	0	NULL
8	BlutdruckMessung	00:15:00	10	1	1	NULL
9	CovidTest	00:30:00	30	2	1	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

Planen von Aufgaben

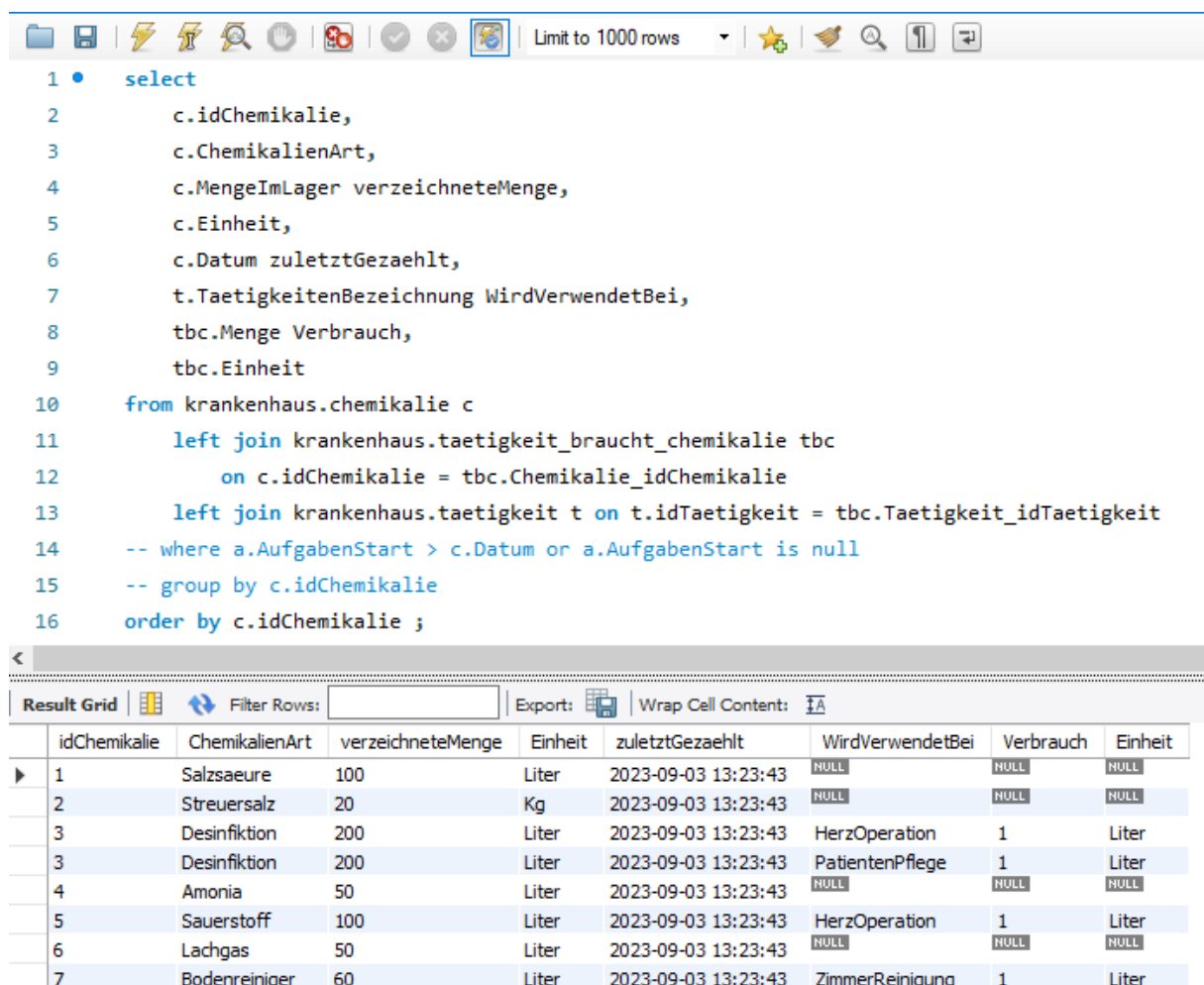
1. Damit eine (Behandlungs-)Aufgabe geplant werden kann, müssen sowohl der Patient als auch die nötige Anzahl Mitarbeiter mit den nötigen Fachkenntnissen, sowie der Raum/Räume mit der nötigen Ausstattung alle gleichzeitig(!) frei sein. Ein komplexer Algorithmus ist erforderlich, um jeweils den frühestmöglichen geeigneten Zeitraum zu finden, alle dafür erforderlichen Daten können jedoch dem **Aufgabenplan** und den verknüpften Tabellen entnommen werden. *(Erforderliche Daten geben Aussagen darüber, wann die Patienten, Mitarbeiter und Räume frei sind. Diese gelten immer dann als frei, wenn sie nicht in einer Aufgabe involviert sind, also nicht im **Aufgabenplan** stehen. Für Mitarbeiter gilt zusätzlich, dass die Aufgabe während ihrer Schicht liegen muss, und nicht während ihrer Pause. Die Pause ist eine Art Aufgabe, die einen Mitarbeiter und einen Pausenraum involviert).*

```
1 • SELECT
2     m.idMitarbeiter,
3     m.Vorname,
4     m.Nachname,
5     m.Stellenbezeichnung,
6     t.TaetigkeitenBezeichnung,
7     a.AufgabenStart,
8     t.Dauer,
9     t.Preis
10  from krankenhaus.mitarbeiter_has_schichtt mhs
11      join krankenhaus.schicht s on s.idSchicht = mhs.Schichtt_idSchicht
12      join krankenhaus.mitarbeiter m on m.idMitarbeiter = mhs.Mitarbeiter_idMitarbeiter
13      join krankenhaus.aufgabenplan_has_mitarbeiter ahm on ahm.Mitarbeiter_idMitarbeiter = m.idMitarbeiter
14      join krankenhaus.aufgabenplan a on a.idAufgabenplan = ahm.Aufgabenplan_idAufgabenplan
15      join krankenhaus.taetigkeit t on t.idTaetigkeit = a.Taetigkeit_idTaetigkeit
16  ;
```

idMitarbeiter	Vorname	Nachname	Stellenbezeichnung	TaetigkeitenBezeichnung	AufgabenStart	Dauer	Preis
2	elcira	weisshaupt	arzt	HerzOperation	2023-09-04 15:00:00	04:00:00	20000
4	herwiga	manger	krankenschwester	HerzOperation	2023-09-04 15:00:00	04:00:00	20000
2	elcira	weisshaupt	arzt	HerzOperation	2023-09-05 15:00:00	04:00:00	20000
4	herwiga	manger	krankenschwester	HerzOperation	2023-09-05 15:00:00	04:00:00	20000
4	herwiga	manger	krankenschwester	BlutdruckMessung	2023-09-04 21:15:00	00:15:00	10
2	elcira	weisshaupt	arzt	BlutdruckMessung	2023-09-04 14:30:00	00:15:00	10

Erfassung des Lagerbestandes

- Der Lagerbestand eines Gutes, beispielsweise eines Medikamentes oder eines Ausstattungsobjektes, wird in der jeweiligen Tabelle, welche die Bezeichnungen für diese Objektkategorie trägt (Bsp.: **Medikamente** oder **Ausstattung**) zusammen mit einer Datumsangabe festgehalten. Das Datum bestimmt den Zeitpunkt, auf den sich der angegebene Lagerbestand bezieht. Um den aktuellen Bestand zu erhalten, muss man von dem angegebenen Bestand alle Mengen abziehen, die seitdem in Aufgaben verbraucht worden sind. Dies ist durch das Verknüpfen von den zugehörigen Tabellen möglich, da jeder Aufgabe eine Tätigkeit zugeordnet ist, und jeder Tätigkeit die Arten und Mengen (mit Einheiten) von benötigten Gütern. Man summiert den Verbrauch aller Aufgaben, um den Gesamtverbrauch zu bekommen. Es ist vorgesehen, dass am Anfang jeden Monats dieser Schritt durchgeführt wird, um den Bestandswert zu aktualisieren.
- Durch Einkäufe erhaltene Güter, können jederzeit problemlos dem Lagerbestand hinzugefügt werden. Momentan werden die einzelnen Einkäufe selbst jedoch nicht festgehalten und Fehler beim Hinzufügen von Mengen zum Lagerbestand können nicht identifiziert werden.

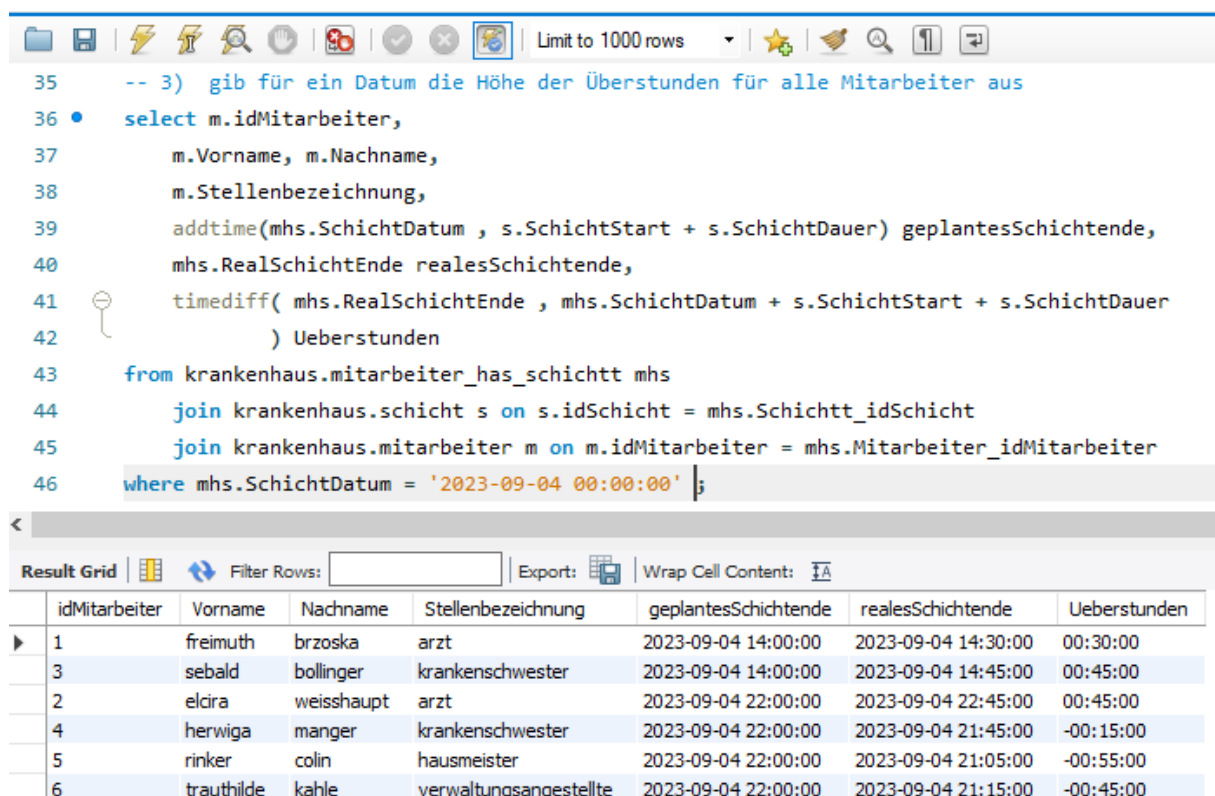


```
1 • select
2     c.idChemikalie,
3     c.ChemikalienArt,
4     c.MengeImLager verzeichneteMenge,
5     c.Einheit,
6     c.Datum zuletztGezaehlt,
7     t.TaetigkeitenBezeichnung WirdVerwendetBei,
8     tbc.Menge Verbrauch,
9     tbc.Einheit
10  from krankenhaus.chemikalie c
11     left join krankenhaus.taetigkeit_braucht_chemikalie tbc
12         on c.idChemikalie = tbc.Chemikalie_idChemikalie
13     left join krankenhaus.taetigkeit t on t.idTaetigkeit = tbc.Taetigkeit_idTaetigkeit
14 -- where a.AufgabenStart > c.Datum or a.AufgabenStart is null
15 -- group by c.idChemikalie
16 order by c.idChemikalie ;
```

	idChemikalie	ChemikalienArt	verzeichneteMenge	Einheit	zuletztGezaehlt	WirdVerwendetBei	Verbrauch	Einheit
1	1	Salzsaure	100	Liter	2023-09-03 13:23:43	NULL	NULL	NULL
2	2	Streuersalz	20	Kg	2023-09-03 13:23:43	NULL	NULL	NULL
3	3	Desinfektion	200	Liter	2023-09-03 13:23:43	HerzOperation	1	Liter
3	3	Desinfektion	200	Liter	2023-09-03 13:23:43	PatientenPflege	1	Liter
4	4	Amonia	50	Liter	2023-09-03 13:23:43	NULL	NULL	NULL
5	5	Sauerstoff	100	Liter	2023-09-03 13:23:43	HerzOperation	1	Liter
6	6	Lachgas	50	Liter	2023-09-03 13:23:43	NULL	NULL	NULL
7	7	Bodenreiniger	60	Liter	2023-09-03 13:23:43	ZimmerReinigung	1	Liter

Schichten, Verspätungen, Überstunden, Gesamtarbeitszeit und Auslastung von Mitarbeitern

1. Die Schicht-Bezeichnung, die -Dauer und die -Startzeit ist für jede der drei Schichtarten jeweils in der Tabelle **Schicht** angegeben. In **Mitarbeiter_hat_Schicht** ist mit jedem Mitarbeiter ein Schicht-Datum und eine Schicht-Bezeichnung verknüpft, so dass man dort nachgucken kann, welche Schicht man an welchem Tag hat.
2. Den geplanten Start der Schicht erhält man durch Schicht-Datum + Schicht-Start, das geplante Ende durch Schicht-Datum + Schicht-Start + Schicht-Dauer
3. In **Mitarbeiter_hat_Schicht** halten die Mitarbeiter auch fest, wann sie die jeweilige Schicht tatsächlich angefangen und beendet haben. Entsprechend ist $\text{TIME}(\text{Schicht-Datum} + \text{Schicht-Start} - \text{Real-Schicht-Start})$ = die Höhe der Verspätung (sofern negativ) und $(\text{Real-Schicht-Ende} - \text{Real-Schicht-Start}) - \text{Schicht-Dauer}$ = die Anzahl der Überstunden für diesen Tag. Die Anzahl der Überstunden kann auch negativ sein und die Höhe der Verspätung auch positiv, sofern ein Mitarbeiter zu früh kommt.
4. $\text{Real-Schicht-Ende} - \text{Real-Schicht-Start}$ = die Anzahl an geleisteten Stunden für diesen Tag. Das zugehörige Datum ist Schicht-Datum oder auch das Datum von Real-Schicht-Start, da es als Datetime festgehalten wird und nicht nur als Time.
5. Die Auslastung eines Mitarbeiters für diesen Tag erhält man, wenn man von den geleisteten Stunden die Zeit abzieht, die mit dem Erledigen von Aufgaben verbracht worden ist. Da jede Aufgabe im Aufgabenplan einen AufgabenStart und ein AufgabenEnde hat, kann man für jede Aufgabe aus den Differenzen von diesen beiden die Dauer bestimmen und über alle Aufgaben aufsummieren.



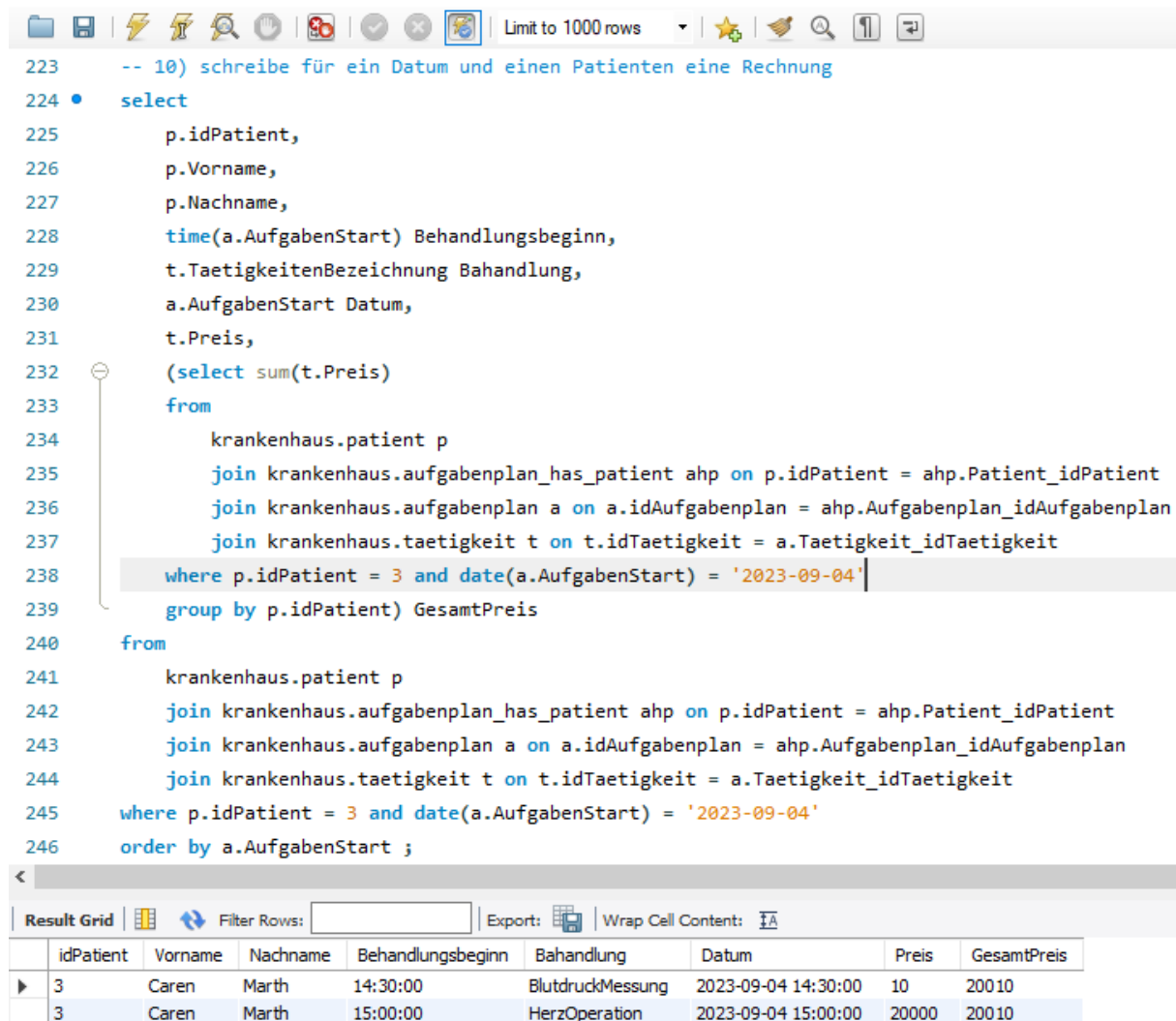
The screenshot shows a SQL IDE interface. At the top, there is a toolbar with various icons and a dropdown menu set to 'Limit to 1000 rows'. Below the toolbar, a SQL query is displayed, starting with a comment '-- 3) gib für ein Datum die Höhe der Überstunden für alle Mitarbeiter aus'. The query uses a series of JOINs to connect the 'mitarbeiter_has_schicht' table with 'schicht' and 'mitarbeiter' tables, and calculates 'geplantesSchichtende', 'realesSchichtende', and 'Ueberstunden' using time functions. The results are shown in a table below the query.

```
35 -- 3) gib für ein Datum die Höhe der Überstunden für alle Mitarbeiter aus
36 • select m.idMitarbeiter,
37        m.Vorname, m.Nachname,
38        m.Stellenbezeichnung,
39        addtime(mhs.SchichtDatum , s.SchichtStart + s.SchichtDauer) geplantesSchichtende,
40        mhs.RealSchichtEnde realesSchichtende,
41        timediff( mhs.RealSchichtEnde , mhs.SchichtDatum + s.SchichtStart + s.SchichtDauer
42                ) Ueberstunden
43 from krankenhaus.mitarbeiter_has_schicht mhs
44     join krankenhaus.schicht s on s.idSchicht = mhs.Schicht_idSchicht
45     join krankenhaus.mitarbeiter m on m.idMitarbeiter = mhs.Mitarbeiter_idMitarbeiter
46 where mhs.SchichtDatum = '2023-09-04 00:00:00' ;
```

	idMitarbeiter	Vorname	Nachname	Stellenbezeichnung	geplantesSchichtende	realesSchichtende	Ueberstunden
▶	1	freimuth	brzoska	arzt	2023-09-04 14:00:00	2023-09-04 14:30:00	00:30:00
	3	sebald	bollinger	krankenschwester	2023-09-04 14:00:00	2023-09-04 14:45:00	00:45:00
	2	elcira	weisshaupt	arzt	2023-09-04 22:00:00	2023-09-04 22:45:00	00:45:00
	4	herwiga	manger	krankenschwester	2023-09-04 22:00:00	2023-09-04 21:45:00	-00:15:00
	5	rinker	colin	hausmeister	2023-09-04 22:00:00	2023-09-04 21:05:00	-00:55:00
	6	trauthilde	kahle	verwaltungsangestellte	2023-09-04 22:00:00	2023-09-04 21:15:00	-00:45:00

Kosten, Umsatz und Gewinn

1. Als Kosten werden nur die Lohnkosten berücksichtigt. Diese sind die Summe der Löhne aller Mitarbeiter (inklusive Überstundenzulage) für diesen Monat
2. Der Umsatz Ist der Gesamtpreis aller durchgeführten Behandlungen. Eine Behandlung ist eine Tätigkeit, die einen Kunden erfordert und einen zugewiesenen Preis hat.
3. Der Gewinn für den jeweiligen Monat ist entsprechend des Umsatzes minus die Kosten.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. The query editor contains a SQL query with line numbers 223 to 246. The query is a SELECT statement that joins patient, task plan, and task tables to calculate the total price for a specific patient on a specific date. The results pane at the bottom shows a table with 8 columns: idPatient, Vorname, Nachname, Behandlungsbeginn, Behandlung, Datum, Preis, and GesamtPreis. Two rows of data are displayed.

```
223 -- 10) schreibe für ein Datum und einen Patienten eine Rechnung
224 • select
225     p.idPatient,
226     p.Vorname,
227     p.Nachname,
228     time(a.AufgabenStart) Behandlungsbeginn,
229     t.TaetigkeitenBezeichnung Behandlung,
230     a.AufgabenStart Datum,
231     t.Preis,
232     (select sum(t.Preis)
233      from
234          krankenhaus.patient p
235          join krankenhaus.aufgabenplan_has_patient ahp on p.idPatient = ahp.Patient_idPatient
236          join krankenhaus.aufgabenplan a on a.idAufgabenplan = ahp.Aufgabenplan_idAufgabenplan
237          join krankenhaus.taetigkeit t on t.idTaetigkeit = a.Taetigkeit_idTaetigkeit
238      where p.idPatient = 3 and date(a.AufgabenStart) = '2023-09-04'
239      group by p.idPatient) GesamtPreis
240 from
241     krankenhaus.patient p
242     join krankenhaus.aufgabenplan_has_patient ahp on p.idPatient = ahp.Patient_idPatient
243     join krankenhaus.aufgabenplan a on a.idAufgabenplan = ahp.Aufgabenplan_idAufgabenplan
244     join krankenhaus.taetigkeit t on t.idTaetigkeit = a.Taetigkeit_idTaetigkeit
245 where p.idPatient = 3 and date(a.AufgabenStart) = '2023-09-04'
246 order by a.AufgabenStart ;
```

	idPatient	Vorname	Nachname	Behandlungsbeginn	Behandlung	Datum	Preis	GesamtPreis
▶	3	Caren	Marth	14:30:00	BlutdruckMessung	2023-09-04 14:30:00	10	20010
	3	Caren	Marth	15:00:00	HerzOperation	2023-09-04 15:00:00	20000	20010

*Beispiele für Ignorierte Daten

1. Nacht- und Feiertagszuschläge
2. Transaktionen für den Einkauf von Gütern
3. Alle Kosten, die keine Lohnkosten sind (Stromkosten, ...)
4. MDH von Medikamenten
5. Verspätungen beim Start von Aufgaben

Fazit

- Auf jede Schätzung der Dauer einer Aufgabe besser 100% drauflegen
- Es ist für die Zusammenarbeit sehr hinderlich, wenn (wie bei der Workbench) an einer Datei nicht mehrere Leute gleichzeitig arbeiten können
- Gute Werkzeuge erleichtern einem die Arbeit unter Umständen erheblich (Zeichenprogramm für ERM)
- Ausfall von Gruppenmitgliedern sollte mit eingeplant werden
- Es ist wichtig sich ein Limit bei dem Umfang der eingeplanten Features zu setzen
- Man sollte darauf vorbereitet sein, unter Umständen seine Freizeit investieren zu müssen
- Trotz erschwerten Bedingungen der individuellen Mitarbeiter, war es uns gemeinsam möglich, ein Projekt auf die Beine zu stellen, da jeder seinen Teil im Rahmen seiner persönlichen Begabung dazu beigetragen hat.

Abgabe Termin von allen Gruppen 06.09.2023 um 11:17, ein Zip Datei, dass alle Dateien beinhaltet. Bitte beachtet, dass alle Daten richtig strukturiert und benannt worden sind.

Folgende Dateien sind erforderlich:

- 1- ER Model als Bild Format
- 2- ER Model (Workbench) als Bild Format
- 3- Dokumentation (Beschreibung, ER-Model, ER-Model (Workbench), Bilder von Tabellen, SQL-Fragen und Antworten mit Bildern, Fazit (was habt Ihr gelernt) ...) – als pdf
- 4- wmb Datei
- 5- Dump Datei – SQL
- 6- CSV von allen Tabellen
- 7- SQL-Fragen und Antworten als txt oder SQL
- 8- Präsentation Datei als opd/PPTX

Benennung von ZIP Datei bitte in folgendes Format:

UKoeln24_[Projektname]_Vornameperson1Nachnameperson1-
Vornameperson2Nachnameperson2 ○

Es wird durch Zufall entschieden welche gruppe als erste präsentiert