

#### 4. Aufgabe (24 Punkte)

Korrekturrand

- a) Sie erhalten den Auftrag, Produktionsdaten an die Steuerung der Walzanlage zu übergeben. Die Produktionsdaten werden in einer SQL-Datenbank gespeichert. Alle Datentypen sind Ganzzahlen. Die Breite, Länge und Dicke der Wellpappe wird in der Datenbank in Millimeter gespeichert.

Die Tabelle ProductionData hat den folgenden Aufbau:

OrderID (PK)
Width
Length
Thickness
Quantity

- aa) Geben Sie den SQL-Befehl an, der die Breite, die Länge, die Dicke und die Anzahl der OrderID 736298 ausgibt. Die OrderID soll nicht in der Ergebnismenge enthalten sein. 3 Punkte

---

---

---

---

- ab) Wie viele Produktionsaufträge für Wellpappen mit einer Dicke von 2 mm wurden bisher in der Datenbank gespeichert. Geben Sie dazu den entsprechenden SQL-Befehl an. 4 Punkte

---

---

---

---

- ac) Geben Sie die Gesamtanzahl gefertigter Wellpappen aus der Datenbank an, die mit einer Dicke von 2 mm, einer Breite von 200 mm und einer Länge von 300 mm gefertigt worden sind. Geben Sie dazu den entsprechenden SQL-Befehl an. 4 Punkte

---

---

---

---

**Dieses Blatt kann an der Perforation aus dem Aufgabensatz herausgetrennt werden!**

## SQL-Syntax

Syntax	Beschreibung
<b>Tabelle</b>	
<b>CREATE TABLE</b> Tabellenname( Spaltenname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
<b>ALTER TABLE</b> Tabellenname <b>ADD COLUMN</b> Spaltenname Datentyp <b>DROP COLUMN</b> Spaltenname Datentyp  <b>ADD FOREIGN KEY</b> (Spaltenname) <b>REFERENCES</b> Tabellenname( Primärschlüsselspaltenname )	Änderungen an einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte  Definiert eine Spalte als Fremdschlüssel
<b>CHARACTER</b>	Textdatentyp
<b>DECIMAL</b>	Numerischer Datentyp (Festkommazahl)
<b>DOUBLE</b>	Numerischer Datentyp (Doppelte Präzision)
<b>INTEGER</b>	Numerischer Datentyp (Ganzzahl)
<b>DATE</b>	Datum (Format DD.MM.YYYY)
<b>PRIMARY KEY</b> (Spaltenname)	Erstellung eines Primärschlüssels
<b>FOREIGN KEY</b> (Spaltenname) <b>REFERENCES</b> Tabellenname( Primärschlüsselspaltenname )	Erstellung einer Fremdschlüssel-Beziehung
<b>DROP TABLE</b> Tabellenname	Löscht eine Tabelle
<b>Befehle, Klauseln, Attribute</b>	
<b>SELECT</b> *   Spaltenname1 [, Spaltenname2, ...]	Wählt die Spalten einer oder mehrerer Tabellen, deren Inhalte in die Liste aufgenommen werden sollen; alle Spalten (*) oder die namentlich aufgeführten
<b>FROM</b>	Name der Tabelle oder Namen der Tabellen, aus denen die Daten der Ausgabe stammen sollen
<b>SELECT ...</b> <b>FROM ...</b> <b>(SELECT ...</b> <b>FROM ...</b> <b>WHERE ...) AS</b> tbl <b>WHERE ...</b>	Unterabfrage (subquery), die in eine äußere Abfrage eingebettet ist. Das Ergebnis der Unterabfrage wird wie eine Tabelle – hier mit Namen "tbl" – behandelt.
<b>SELECT DISTINCT</b>	Eliminiert Redundanzen, die in einer Tabellen auftreten können, Werte werden jeweils nur einmal angezeigt.
<b>JOIN / INNER JOIN</b>	Liefert nur die Datensätze zweier Tabellen, die gleiche Datenwerte enthalten
<b>LEFT JOIN / LEFT OUTER JOIN</b>	Liefert von der erstgenannten (linken) Tabelle alle Datensätze und von der zweiten Tabelle jene, deren Datenwerte mit denen der ersten Tabelle übereinstimmen
<b>RIGHT JOIN / RIGHT OUTER JOIN</b>	Liefert von der zweiten (rechten) Tabelle alle Datensätze und von der ersten Tabelle jene, deren Datenwerte mit denen der zweiten Tabelle übereinstimmen
<b>WHERE</b>	Bedingung, nach der Datensätze ausgewählt werden sollen
<b>WHERE EXISTS</b> ( subquery ) <b>WHERE NOT EXISTS</b> ( subquery )	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
<b>WHERE ... IN</b> ( subquery ) <b>WHERE NOT... IN</b> ( subquery )	Der Wert des Datenfeldes ist in der ausgewählten Menge vorhanden. Der Wert des Datenfeldes ist in der ausgewählten Menge nicht vorhanden.
<b>GROUP BY</b> Spaltenname1 [,Spaltenname2, ...]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
<b>ORDER BY</b> Spaltenname1 [,Spaltenname2, ...] <b>ASC   DESC</b>	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend

Fortsetzung SQL-Syntax →

Syntax	Beschreibung
<b>Datenmanipulation</b>	
<b>DELETE FROM</b> Tabellenname	Löschen von Datensätzen in der genannten Tabelle
<b>UPDATE</b> Tabellenname <b>SET</b>	Aktualisiert Daten in Feldern einer Tabelle
<b>INSERT INTO</b> Tabellenname[(spalte1, spalte2, ...)] <b>VALUES</b> (Wert für Spalte 1 [, Wert für Spalte 2, ...]) oder <b>SELECT ... FROM ... WHERE</b>	Fügt Datensätze in die genannte Tabelle, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind
<b>Berechtigungen kontrollieren</b>	
<b>CREATE</b> Benutzer   Rolle <b>IDENTIFIED BY</b> 'Passwort'	Erzeugt einen neuen Benutzer oder eine neue Rolle mit einem Passwort
<b>GRANT</b> Recht   Rolle <b>ON</b> *.*   Datenbank.*   Datenbank.Objekt <b>TO</b> Benutzer   Rolle [WITH GRANT OPTION]	Weist einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt zu Weist einem Benutzer eine Rolle zu
<b>REVOKE</b> Rechte   Rollen <b>ON</b> *.*   Datenbank.*   Datenbank.Objekt <b>FROM</b> Benutzer   Rolle	Entzieht einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt Entzieht einem Benutzer eine Rolle
<b>Aggregatfunktionen</b>	
<b>AVG</b> (Spaltenname)	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
<b>COUNT</b> (Spaltenname   * )	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
<b>SUM</b> (Spaltenname   Formel)	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
<b>MIN</b> (Spaltenname   Formel)	Ermittelt den kleinsten aller Werte im angegebenen Feld
<b>MAX</b> (Spaltenname   Formel)	Ermittelt den größten aller Werte im angegebenen Feld
<b>Funktionen</b>	
<b>LEFT</b> (Zeichenkette, Anzahlzeichen)	Liefert <i>Anzahlzeichen</i> der Zeichenkette von links.
<b>RIGHT</b> (Zeichenkette, Anzahlzeichen)	Liefert <i>Anzahlzeichen</i> der Zeichenkette von rechts.
<b>CURRENT</b>	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
<b>CONVERT</b> (time,[DatumZeit])	Liefert die Uhrzeit aus einer DatumZeit-Angabe
<b>DATE</b> (Wert)	Wandelt einen Wert in ein Datum um
<b>DAY</b> (Datum)	Liefert den Tag des Monats aus dem angegebenen Datum
<b>MONTH</b> (Datum)	Liefert den Monat aus dem angegebenen Datum
<b>TODAY</b>	Liefert das aktuelle Datum
<b>WEEKDAY</b> (Datum)	Liefert den Tag der Woche aus dem angegebenen Datum
<b>YEAR</b> (Datum)	Liefert das Jahr aus dem angegebenen Datum
<b>DATEADD</b> (Datumsteil, Intervall, Datum)	Fügt einem Datum ein Intervall (ausgedrückt in den unter Datumsteil angegebenen Einheiten) hinzu
<b>DATEDIFF</b> (Datumsteil, Anfangsdatum, Enddatum) Datumsteile: <b>DAY, MONTH, YEAR</b>	Liefert Enddatum-Startdatum (ausgedrückt in den unter Datumsteil angegebenen Einheiten)
<b>Operatoren</b>	
<b>AND</b>	Logisches UND
<b>LIKE</b>	Überprüfung von Text auf Gleichheit wenn Platzhalter ("regular expressions") eingesetzt werden.
<b>NOT</b>	Logische Negation
<b>OR</b>	Logisches ODER
<b>IS NULL</b>	Überprüfung auf NULL
<b>=</b>	Test auf Gleichheit
<b>&gt;, &gt;=, &lt;, &lt;=, &lt; &gt;</b>	Test auf Ungleichheit
<b>*</b>	Multiplikation
<b>/</b>	Division
<b>+</b>	Addition, positives Vorzeichen
<b>-</b>	Subtraktion, negatives Vorzeichen

Stand 2021-09-30

- b) Die abgefragten Produktionsdaten werden über eine entsprechende API an die Steuerung der Walzanlage übergeben. Die Auftragsdaten werden im Array `result[]` mit dem Index 0 bis 3 gespeichert. Sie sollen jetzt an die Steuerung der Walzanlage durch eine von Ihnen zu erstellende Funktion übergeben werden. Gehen Sie von einem Array `result[]` aus, bei dem im Index 0 die Breite, im Index 1 die Länge, im Index 2 die Dicke und im Index 3 die Anzahl der zu produzierenden Wellpappen stehen.

Erstellen Sie die Funktion „**launchTask(result[])**“.

Zur Kommunikation mit der Steuerung der Walzanlage stehen Ihnen die folgenden API-Funktionen zur Verfügung:

**setRollerDim(int,int,int)** – Übergeben wird Breite, Länge und Dicke der Wellpappe.

**rollerStart()** – Startet einen Auftrag von einem Stück. Es wird eine Wellpappe mit den gesetzten Parametern erzeugt.

Die Walzanlage verfügt über einen Notauschalter. Sie darf nur laufen, wenn der Notaus nicht ausgelöst ist.

Der Status des Notauschalters kann mit der Funktion **bool getEmergencyStop()** abgefragt werden, der „true“ liefert wenn der Notaus ausgelöst ist und „false“ wenn der Notaus nicht ausgelöst ist.

Ergänzen Sie das gegebene Struktogramm durch die entsprechenden Befehle zur Produktion der geforderten Anzahl von Wellpappen (siehe Index 3) in den angegebenen Maßen (siehe Index 0, 1 und 2). 7 Punkte

<b>launchTask(result[])</b>	
<b>int i = 0</b>	
<b>bool emergencyStop = getEmergencyStop()</b>	



#### Fortsetzung 4. Aufgabe

Korrekturrand

- c) Für die Produktion von Wellpappen ist die vorhandene Datenbank zu erweitern. Die Firma hat sich für ein SQL-fähiges relationales Datenbanksystem entschieden, in der die nachfolgenden Bedingungen berücksichtigt werden sollen. Die Speicherung der Datenbank wird auf dem Hostrechner „Steuerungs-PC“ realisiert. In einer ersten Unterredung werden die zu speichernden Informationen definiert.

In dieser Datenbank sollen nur die Zusammenhänge zwischen den Walzanlagen, den Produktionsdaten abgebildet werden.

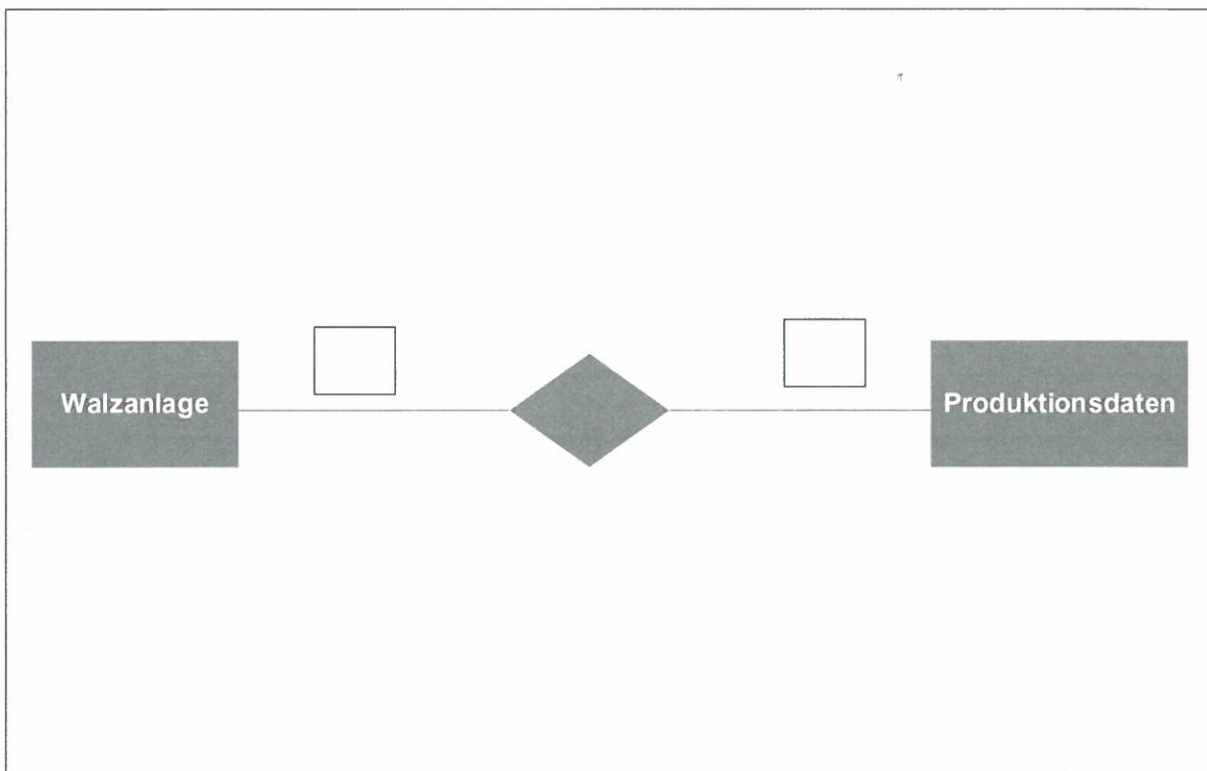
In der Produktionshalle sind mehrere Walzanlagen vorhanden. Diese jeweiligen Walzanlagen können Wellpappen mit unterschiedlichen Dicken (z. B. kleiner 4 mm, 4-8 mm, 8-12 mm) herstellen. In der Datenbank soll gespeichert werden, welche Walzanlage für welche Dicken (Spezifikation) verwendet werden kann. Außerdem soll das Baujahr, die Bezeichnung und eine eindeutige Maschinennummer gespeichert werden.

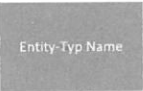


Für jede Walzanlage sollen die entsprechenden Produktionsdaten (Breite, Länge, Dicke und Anzahl) mit dem jeweiligen Zeitstempel abgespeichert werden.

Vervollständigen Sie das vorgegebene Entity-Relationship-Modell (kurz: ERM) für diese Datenbank mit allen erforderlichen Attributen und Kardinalitäten.

6 Punkte

Hinweis: Die eventuell benötigten Fremdschlüssel müssen nicht in diesem Entwurf eingetragen werden. Die Kardinalität zwischen den beiden Tabellen soll auf die entsprechenden Beziehungslinien eingetragen werden.



Bezeichnung	Darstellung
Entity-Typ	
Attribut	
Primärschlüssel	
Beziehung (Relation, Relationship, Assoziation)	

**PK** bezeichnet ein Primärschlüsselattribut, **FK** ein Fremdschlüsselattribut, Primärschlüsselattribute werden unterstrichen, Fremdschlüsselattribute werden durch ein nachgestelltes Hash-Zeichen (#) kenntlich gemacht.

**PRÜFUNGSZEIT – NICHT BESTANDTEIL DER PRÜFUNG!**

Wie beurteilen Sie nach der Bearbeitung der Aufgaben die zur Verfügung stehende Prüfungszeit?

- ☐ 1 Sie hätte kürzer sein können.
- ☐ 2 Sie war angemessen.
- ☐ 3 Sie hätte länger sein müssen.