



WFF-Datenbank

Annemarie Lebbink

DLR

Cluster:	3 (Turnaround)
Dokument Nr.:	D3126
Version Nr.:	1.00
Einstufung:	öffentlich
Anzahl der Seiten:	29

Projekt Manager

Dr. Jens Konopka

Deutsche Flugsicherung GmbH

63202 Langen, Deutschland

Telefon: +49 (0)6103 707 5792, Fax: +49 (0)6103 707 5741

Web page:

Dokumentenkontrolle

Projekt Manager	Dr. Jens Konopka	
Verantwortlicher Autor	Annemarie Lebbink	DLR
Weitere Autoren	Friedrich Posnien	DLR
	Steffen Loth	DLR
Titel des Dokumentes:	WFF-Datenbank	
Zugehöriges Arbeitspaket:	AP 312	
Dokument Nr.	D3126	
Speicherdatum:	2010-03-31	
Dokument Version:	1.00	
Datei Name	D3126_WFF_Datenbank_V100.doc	
Anzahl der Seiten	29	
Status Geheimhaltung	öffentlich	

Änderungsverfolgung (Change Log)

Datum	Version	Änderungen Kapitel/Absatz	Kommentar
03.02.2009	0.01	Initial Draft	
08.02.2009	0.02	Überarbeitung der Version 0.01	
12.03.2010	0.03	Korrektur der Version 0.02	
16.03.2010	0.04	Final Review	
31.03.2010	1.00	Endversion	

Inhalt

Dokumentenkontrolle	2
Änderungsverfolgung (Change Log)	2
Inhalt	3
1 Ziel des Dokumentes	4
2 Datenbanksoftware	5
3 Datenbankmodell	6
4 Datenbankstruktur	11
4.1 Tabelle tdt_flight	12
4.2 Tabellen tdt_departure / tdt_arrival	14
4.3 Tabelle tadd_aircraft_type	15
4.4 Tabelle tadd_position	16
4.5 Tabelle tadd_luggage_belt	17
4.6 Tabelle tlhk_turnaround	18
4.7 Tabelle tadd_pax_flight	19
4.8 Tabelle tdt_sita_messages	19
4.9 Tabelle tdt_task_luggage	20
4.10 Tabelle tdt_task_part_luggage_pax	21
4.11 Tabelle tdt_task_part_luggage_abort	23
4.12 Tabelle tdt_task_part_luggage_messages	24
4.13 Tabelle tdt_vehicle	25
4.14 Tabelle tdt_luggagecar	26
4.15 Tabelle tdt_employee	27
4.16 Tabelle ttmp_updates_dlr Tabelle / tttmp_updates_uhh	27
5 Anhang I	28
5.1 Abkürzungsverzeichnis / Glossar	28
5.2 Abbildungsverzeichnis	28
5.3 Tabellenverzeichnis	28

1 Ziel des Dokumentes

Zur Umsetzung des Fahrzeugmanagementsystems wurde eine Datenbank benötigt, die zum einen Daten des real am Flughafen stattfindenden Verkehrs enthält und zum anderen die Möglichkeit bietet Planungsdaten zum Einsatz der Ressourcen abzulegen und die Kommunikation zwischen den Teilsystemen erlaubt. Neben der Nutzung für die Durchführung der Einsatzplanung und Auftragsübermittlung der Gepäckfahrzeuge stellt die Datenbank auch das zentrale Verbindungsglied zwischen Optimierung und HMI dar.

Das vorliegende Dokument beschreibt den Aufbau und die Inhalte der im Rahmen des Projektes erstellten Datenbank. Es geht dabei detailliert auf das Datenbankmodell, die Relationen und die Tabellen ein.

Das Dokument ist wie folgt gegliedert:

- Kapitel 1 Ziel des Dokumentes (dieses Kapitel)
- Kapitel 2 Datenbanksoftware
- Kapitel 3 Datenbankmodell
- Kapitel 4 Datenbankstruktur
- Kapitel 5 Verzeichnisse

2 Datenbanksoftware

Im Rahmen des WFF Projektes wurde als Datenbankmodell MySQL 5.1.30 gewählt. Dies hat folgende Gründe:

Als Open-Source-Produkt mit GNU General Public License ist MySQL kostenlos.

MySQL ist ein relationales Datenbankmanagementsystem und plattformunabhängig einsetzbar. Eine relationale Datenbank speichert Daten in separaten Tabellen, statt alle Daten in einem einzigen großen Speicherraum abzulegen. Das sorgt für Vorteile in Bezug auf Geschwindigkeit und Flexibilität. Die Tabellen können über Schlüssel, die zur eindeutigen Identifizierung dienen, miteinander verknüpft werden.

Die MySQL-Software ist ein schneller und stabiler SQL-Datenbankserver (Structured Query Language, strukturierte Abfragesprache) und funktioniert als Client-Server-System.

MySQL ist leicht installierbar und benutzerfreundlich.

Bei Version 5.1 wird die Speicher-Engine InnoDB als Standard-Engine installiert, entscheidender Vorteil dieser Engine ist die Transaktionssicherheit und die Gewährleistung der referenziellen Integrität über Fremdschlüssel.

Für die Konfiguration des MySQL Servers sollten folgende Einstellungen in den jeweiligen Dialogfenstern gewählt werden:

Configuration type:	Detailed Configuration
Server type:	Developer Machine
Database usage:	Multifunctional Database
Concurrent connections:	Decision Support (DSS)/OLAP
Networking options:	Enable TCP/IP Networking, Port Number: 3306, Enable strict Mode
Default character set:	Standard Character Set
Windows options:	Install As Windows Service
Security options:	Modify Security Settings

Der MySQL Server wurde im Forschungsnetzwerk der Testplattform aufgesetzt. Er verwaltet die Datenbank, auf die von allen anderen Rechnern zugegriffen werden kann. Zur Wartung der Datenbank wurden verschiedene Tools eingesetzt, die über grafische Oberflächen einen einfachen Zugriff auf Strukturen und Inhalte der Datenbank ermöglichen.

3 Datenbankmodell

Ein Datenbankmodell repräsentiert einen Ausschnitt aus der Realwelt. Der Realitätsausschnitt für den Groundhandling Manager ist der Arbeitsablauf der Gepäcktransportabwicklung in der Einsatzzentrale am Hamburger Flughafen.

Als wesentliche Prozesse der Gepäcktransportabwicklung können unterschieden werden:

- Anlieferung von leeren trolleys (Gepäckanhänger) zum Flugzeug (arrival)
- Abholung der mit Gepäckstücken beladener trolleys vom Flugzeug (arrival)
- Anlieferung der mit Gepäckstücken beladener trolleys vom Flugzeug (departure)
- Abholung von leeren trolleys vom Flugzeug (departure)

Zur Durchführung der Einsatzplanung und Auftragsübermittlung von Gepäckfahrzeugen werden die notwendigen Prozesse in einer Datenbank abgebildet.

So müssen zum Beispiel für die Abbildung des Prozesses der Gepäcktransportabwicklung alle zur Verfügung stehenden Gepäckfahrzeuge, die zugehörigen Gepäckfahrzeugnummern und die Gepäckfahrzeugpositionen hinterlegt werden. Neben diesen Daten sollen auch die zugehörigen Informationen über die, den Fahrzeugen zugeordneten Fahrer und der Status der Gepäckfahrzeuge hinterlegbar sein. Zusätzlich müssen die aktuellen Flugpläne mit spezifischen Flugplandaten wie Flugnummer, Flugzeugtyp, Position, SIBT, EIBT, SOBT, EOBT etc. in der Datenbank verfügbar sein. Aus diesen Daten können dann die Gepäckfahraufträge für einzelne Flüge generiert werden.

Die Planungs- und Auftragsdaten eines Managementsystems der Universität Hamburg treiben das Management HMI des DLR und die Daten der damit erstellten Fahraufträge werden zu diesem Managementsystem zurück geführt. Diese Daten müssen daher auch in der Datenbank hinterlegbar sein. Neben den technischen Informationen muss das Datenbankmodell also zusätzlich in der Lage sein Informationen von anderen und für andere Systeme aufnehmen zu können.

Als Werkzeug zur konzeptionellen Datenmodellierung wird auf das Entity-Relationship Modell zurückgegriffen. Die Objekte werden dabei in Entities umgesetzt, die bestimmte Eigenschaften (Attributes) besitzen und miteinander in Beziehung stehen (Relationship). Jedes Objekt und jede Beziehung zwischen den Objekten entspricht einer zweidimensionalen Tabelle:

Jede Spalte einer Tabelle steht für ein Attribut des Objektes. Die Spalte muss innerhalb der Tabelle eindeutig sein. Die Anordnung der Spalten innerhalb der Tabelle ist beliebig.

Jede Zeile enthält jeweils alle Attribute, die ein Objekt oder eine Beziehung charakterisieren. In einer Tabelle gibt es keine identischen Zeilen, jede Zeile ist eindeutig. Ein Schlüsselattribut dient zur eindeutigen Identifizierung. Die Anordnung der Zeilen innerhalb der Tabelle ist beliebig.

Eine komplette Zeile einer Datenbanktabelle wird als Datensatz betrachtet.

Ein Feld ist eine bestimmte Spalte aus einem Datensatz.

Zum besseren Verständnis und zur Vereinfachung der Beschreibung der verwendeten Datenbankstruktur und der Tabellenrelationen dient die grafische Darstellung des Datenbankmodells in Abbildung 1.

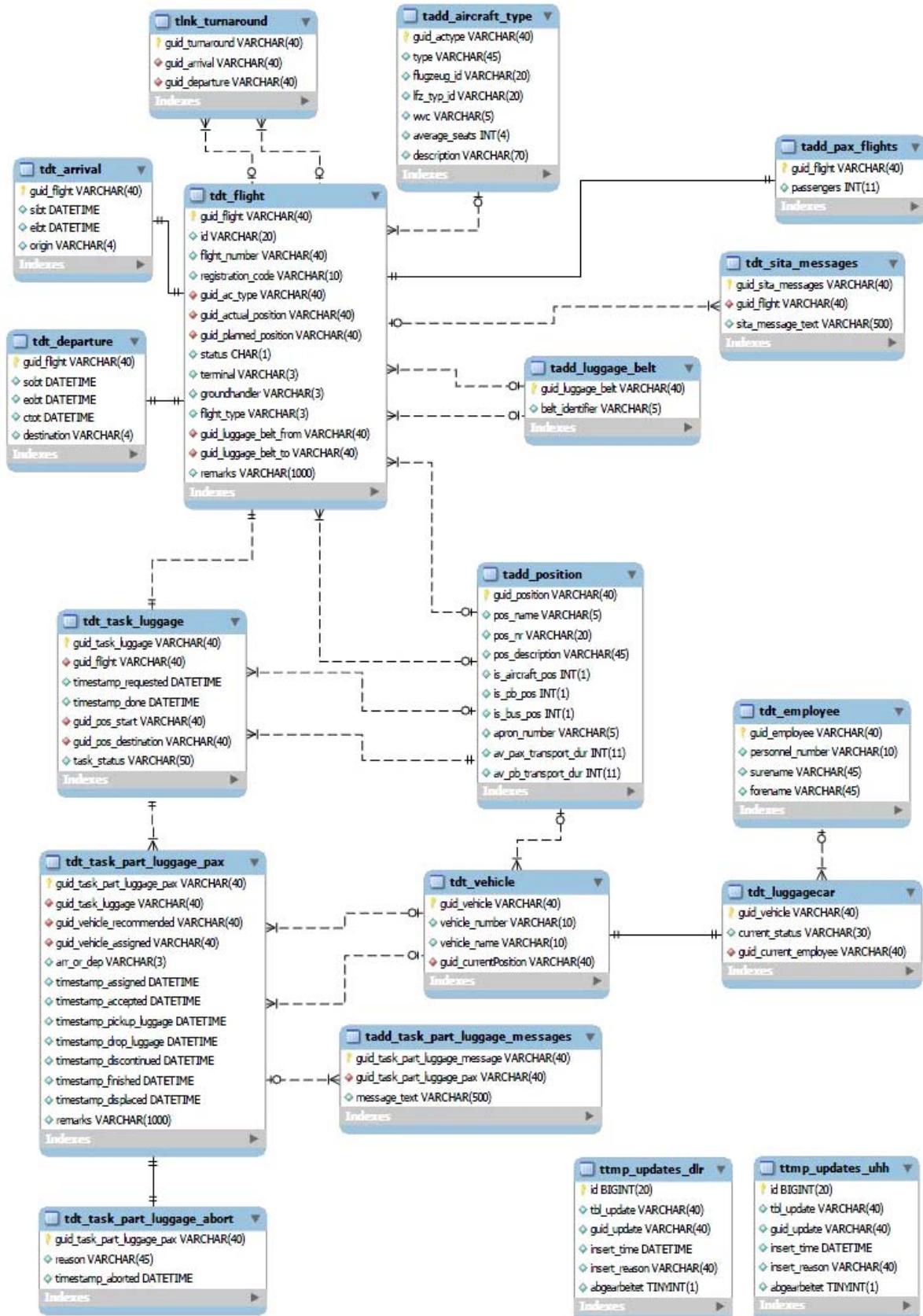


Abbildung 1: Darstellung des Datenbankmodells

In Abbildung 1 wird deutlich, dass alle Tabellen außer den Austauschtabellen tttmp_updates_dlr und tttmp_updates_dlr miteinander verknüpft sind. In dem hier beschriebenen Datenbankmodell sind folgende Typen von Verknüpfungen relevant:

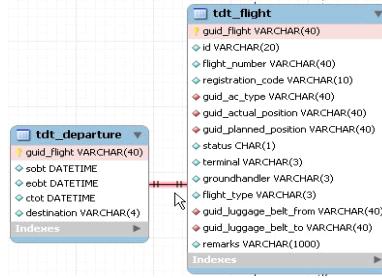
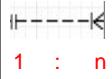
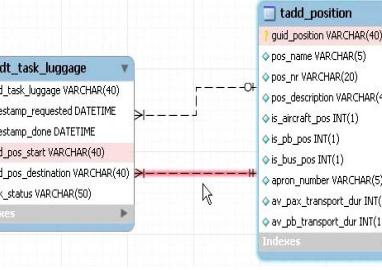
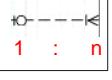
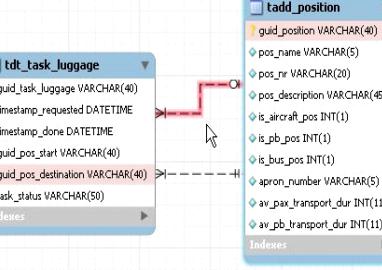
Beziehung	Symbol	Beispiel	Beschreibung
1:1	 1 : 1	<p>Referenced Tabel Mandatory: on Referencing Tabel Mandatory: on</p> 	<p>Bei einer 1:1-Beziehung ist jeder Datensatz in einer Tabelle (z.B. tdt_flight) direkt mit einem einzigen Datensatz in einer anderen Tabelle (z.B. tdt_departure) verknüpft: Da jeder Flug entweder nur ein arrival ODER ein departure sein kann, ist dies eine 1:1 Beziehung. Somit ist der Primärschlüssel guid_flight in tdt_flight als Feld der Tabelle tdt_departure hinzuzufügen.</p>
1:n	 1 : n	<p>Referenced Tabel Mandatory: on Referencing Tabel Mandatory: on</p> 	<p>Bei einer 1:n-Beziehung kann ein Datensatz einer Tabelle (z.B. tadd_position) mit beliebig vielen Datensätzen einer anderen Tabelle (z.B. tdt_task_luggage) verbunden sein, aber nicht umgekehrt. (Bsp: jeder Gepäckauftrag hat nur einen Startpunkt, aber ein und derselbe Startpunkt kann in mehreren Gepäcksaufrägen auftauchen). Weiterhin gilt eine referentielle Integrität. Sprich MySQL überprüft, ob alle Werte des Fremdschlüsselfeldes der Tabelle auf der n-Seite der Beziehung (z.B. tdt_task_luggage) auch in der verknüpften Tabelle als Primärschlüssel enthalten sind. Zusätzlich gilt noch, dass im Fremdschlüsselfeld auf der n-Seite der Beziehung ein NULL-Eintrag als Wert NICHT zulässig ist (Bsp: guid_pos_start ≠ NULL in tdt_task_luggage)</p>
1:n Null zulässig	 1 : n	<p>Referenced Tabel Mandatory: off Referencing Tabel Mandatory: on</p> 	<p>Enspricht einer 1:n Beziehung allerdings mit folgender Einschränkung Im Fremdschlüsselfeld auf der n-Seite der Beziehung ist der NULL-Eintrag als Wert zulässig ist (Bsp: guid_pos_destination = NULL in tdt_task_luggage)</p>

Tabelle 1: Verknüpfungen zwischen den einzelnen Tabellen

Die Verknüpfung entsteht durch die Verwendung von Primärschlüsseln:

- Der Primärschlüssel des Datensatzes einer Tabelle wird als Wert in einem Feld einer anderen Tabelle eingetragen.
- Der Primärschlüssel einer Tabelle, auf den in einer anderen Tabelle verwiesen wird, heißt dort Fremdschlüssel.
- Der Primärschlüssel hat innerhalb der Tabelle einen einmaligen Wert und garantiert somit die Eindeutigkeit des Datensatzes.
- Kein Primärschlüssel darf NULL sein.

Abbildung 2 stellt ein Beispiel für einen Primär- und Fremdschlüssel dar:

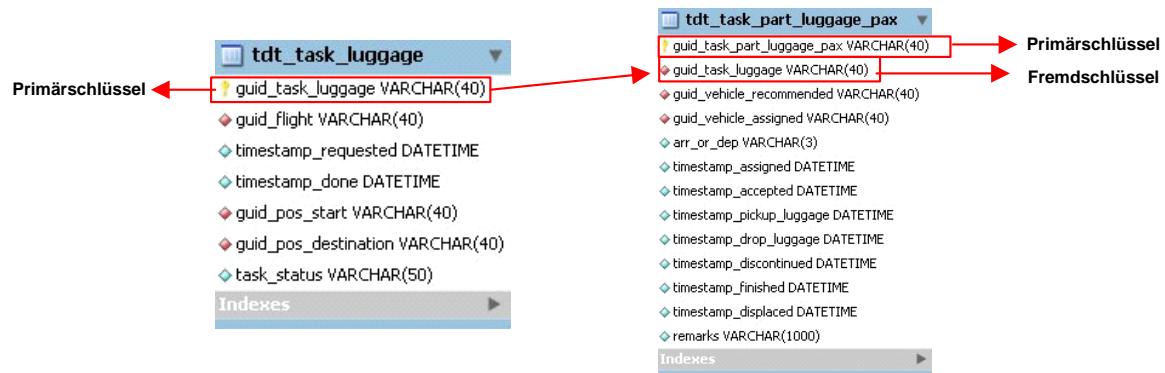


Abbildung 2: Primärschlüssel/Fremdschlüssel in einer Tabelle

In der Tabelle `tdt_task_luggage` ist `guid_task_luggage` der Primärschlüssel, in der Tabelle `tdt_task_part_luggage_pax` ist es der `guid_task_part_luggage_pax`. Die Tabelle `tdt_task_luggage` ist die Mastertabelle oder auch Parent-Tabelle genannt, die Tabelle `tdt_task_part_luggage_pax` ist die Child-Tabelle. Die Information, zu welchem task luggage welcher task part luggage pax gehört, wird durch die Spalte `guid_task_luggage` festgehalten. Der Primärschlüssel in `tdt_task_luggage` wird in der `tdt_task_part_luggage_pax` Tabelle als Fremdschlüssel verwendet.

In diesem Datenbankmodell werden als Primärschlüssel GUIDs mit 128 Bit verwendet, die das Format (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx) mit 32 Zeichen haben, wobei jedes x entweder eine Ziffer von 0-9 oder ein Buchstabe von A-F sein kann.

Die beiden Austauschtabellen `ttmp_updates_dlr` und `ttmp_updates_uhh` haben keine Beziehung zueinander und zu anderen Tabellen, da sie autarke Daten enthalten, die aber unter anderem für die Kommunikation zwischen dem Management-HMI und dem Managementsystem der Universität Hamburg erforderlich sind. Die für das Management-HMI relevanten Daten, die vom Managementsystem der Universität Hamburg gesendet werden, werden in die `ttmp_updates_dlr` Tabelle eingetragen. Umgekehrt werden in der `ttmp_updates_uhh` Tabelle die für das Managementsystem der Universität Hamburg relevanten Daten, die vom Management-HMI gesendet werden, eingetragen.



Abbildung 3: Austauschtabellen

Eine Quittierung der Übertragung erfolgt nicht, so dass auf Senderseite nie klar ist, ob die gesendeten Daten auch wirklich beim Empfänger ankommen sind.

Jeder Eintrag in eine der beiden Tabellen hat eine Sequenznummer, die dann inkrementiert wird. Diese Nummer ist eindeutig und wird als Primärschlüssel verwendet.

Beim Management HMI wird zu Beginn des Programmstarts ein Timer angestoßen, der sekündlich überprüft, ob neue Nachrichten in der `ttmp_updates_dlr` Tabelle eingetroffen sind. Neue Nachrichten sind all diejenigen Datensätze, bei denen im Feld „`abgearbeitet`“ ein 0-Eintrag vorhanden ist. Das Programm hat die Aufgabe, alle „`abgearbeitet = 0`“ Datensätze anzunehmen, abzuarbeiten und bei erfolgreicher Abarbeitung zu quittieren. Die Quittung bei erfolgreicher Abarbeitung besteht aus dem Datenbankeintrag „`abgearbeitet = 1`“. Unvollständig oder falsch gesendete Datensätze werden beim Einlesen erkannt und führen zum Programmabsturz. Diese Datensätze müssen aus Datenbank gelöscht werden, damit ein Programmstart wieder möglich ist.

4 Datenbankstruktur

Die Datenbank ist aus Tabellen aufgebaut. Folgende Typen von Tabellen wurden definiert:

Präfix	Tabellentyp	Beschreibung
tdt	Datentabelle	Haupttabelle, enthält die Kerndaten des Objektes.
tadd	Ergänzende Tabelle	Nebentabelle, enthält Daten, die Detailinformationen zu den Feldern in der Datentabelle darstellen.
tlnk	Verknüpfungstabelle	Tabelle zum Verbinden zweier Tabellen mit einer m:n-Beziehung.
ttmp	Temporäre Tabelle	Tabelle, die zur temporären Zwischenspeicherung von Daten und zur Kommunikation mit den externen Systemen (A-SMGCS, Optimierer) genutzt wird. Daten in dieser Tabelle können jederzeit gelöscht werden

Tabelle 2: Tabellentypen

Im Folgenden werden die im Datenbankmodell zur Darstellung und Durchführung der Einsatzplanung und Auftragsübermittlung von Gepäckfahrzeugen erstellten Tabellen beschrieben.

Die Zeilen der jeweiligen Tabellen stellen die einzelnen Attribute dar, die das Objekt in der jeweiligen Datenbanktabelle charakterisieren. Die Spalten beschreiben das jeweilige Attribut hinsichtlich:

- des Datentyps
- der Variablenlänge
- der Eigenschaft, ob es sich um einen Primärschlüssel oder einen Fremdschlüssel handelt und ob ein Dateneintrag Pflicht oder optional ist
- der Referenz zu einer anderen Tabellen, sofern eine Referenz existiert

Zum besseren Verständnis erfolgt in der Spalte „Beschreibung“ eine ausführliche Beschreibung des Attributes

4.1 Tabelle tdt_flight

Das Management HMI stellt die aktiven Flugpläne für arrivals und departures tabellarisch dar. Die zentrale Funktionalität der Flugplantabellen besteht neben der Sichtungsmöglichkeit der aktuellen Flugpläne darin, Fahraufträge für einzelne Flugpläne generieren zu können. Die dargestellten Flugplandaten werden aus der Datenbank ausgelesen. Die Tabelle tdt_flight ist die Haupttabelle für die Flugplandaten. Die Struktur ist so angelegt, dass die Flugplandaten, die unabhängig der departure oder arrival Spezifizierung sind, wie z.B. Flugnummer, Registrierungscode, Flugzeugtype, aktueller Standplatz, geplanter Standplatz, Status, Terminal, Gewichtsklasse, von Gepäckband, nach Gepäckband etc. direkt in der Tabelle tdt_flight abgespeichert werden. Die guid_flight ist der Primärschlüssel.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_flight	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für Flug (GUID)
id	varchar	20	Eintrag Pflicht		Der in der Hamburger Flughafendatenbank FB2000 verwendeter Identifier für den Flug
flight_number	varchar	40	Eintrag Pflicht		Callsign
registration_code	varchar	10	Eintrag optional		Registrierungscode
guid_ac_type	varchar	40	Fremdschlüssel, Eintrag optional	Verweis auf Tabelle tadd_aircraft_type	Vierstelliger Code für den Typ des Flugzeugs (Bsp. A345)
guid_actual_position	varchar	40	Fremdschlüssel, Eintrag optional	Verweis auf Tabelle tadd_position (guid_position)	Aktueller Standplatz
guid_planned_position	varchar	40	Fremdschlüssel, Eintrag optional	Verweis auf Tabelle tadd_position (guid_position)	Geplanter Standplatz
status	char	1	Eintrag optional		Einstelliger Eintrag zur Darstellung des aktuellen Status des Flugzeugs: z.B. 4 = 40 miles zum touch down, N = on-block, L = gelandet))
terminal	varchar	3	Eintrag optional		Gepäckterminal
groundhandler	varchar	3	Eintrag optional		Name des Gepäckbodenabfertigungsdienst
flight_type	varchar	3	Eintrag optional		Flugzeugklassifizierung, wie z.B. J = Jet, PR = Prop etc.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_luggage_belt_from	varchar	40	Fremdschlüssel, Eintrag optional	Verweis auf Tabelle tadd_luggage_belt (guid_luggage_belt)	Vom Gepäckband
guid_luggage_belt_to	varchar	40	Fremdschlüssel, Eintrag optional	Verweis auf Tabelle tadd_luggage_belt (guid_luggage_belt)	Zum Gepäckband
remarks	varchar	1000	Eintrag optional		Zum Flug zusätzliche für den Groundhandling Disponent relevanten Informationen

Tabelle 3: Tabelle tdt_flight

Durch die Spezifizierung des Fluges als arrival oder departure, werden der tdt_flight Tabelle über die beiden Tabellen tdt_departure und tdt_arrival zusätzliche Informationen zugewiesen. Tabelle 4 und Tabelle 5 beschreiben den Aufbau dieser Tabellen.

Die tdt_flight Tabelle wird jeweils:

- mit dem Feld guid_ac_type der Tabelle tadd_aircraft_type
- mit dem Feld guid_actual_position der Tabelle tadd_position
- mit dem Feld guid_planned_position der Tabelle tadd_position
- mit dem Feld guid_lugage_belt_from der Tabelle tadd_luggage_belt
- mit dem Feld guid_luggage_belt_to der Tabelle tadd_luggage_belt

verknüpft.

Diese Tabellen sind ergänzende Tabellen zur tdt_flight Tabelle. Tabelle 6, Tabelle 7 und Tabelle 8 stellen jeweils den Aufbau dieser Nebentabellen dar.

4.2 Tabellen tdt_departure / tdt_arrival

Jeder Flug wird in der Datenbank entweder als ein departure oder ein arrival geführt. Seinem Typ entsprechend wird er zusätzlich mit spezifischen Flugplandaten versehen. Diese Daten werden über eine 1:1 Verknüpfung in der Tabelle tdt_arrival bzw. tdt_departure gespeichert. Die Verknüpfung erfolgt über die guid_flight, welche bei der Eingabe der Stammdaten für jeden Datensatz automatisch vergeben wird.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_flight	varchar	40	Primärschlüssel, Eintrag Pflicht	Verweis auf Tabelle tdt_flight (guid_flight)	Identifier für Flug (GUID)
sibt	datetime		Eintrag optional		Scheduled InBlock Time
eibt	datetime		Eintrag optional		Estimated InBlock Time
origin	varchar	4	Eintrag optional		Startflughafen

Tabelle 4: Tabelle tdt_arrival

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_flight	varchar	40	Primärschlüssel, Eintrag Pflicht	Verweis auf Tabelle tdt_flight (guid_flight)	Identifier für Flug (GUID)
sobt	datetime		Eintrag optional		Scheduled OffBlock Time
eobt	datetime		Eintrag optional		Estimated OffBlock Time
ctot	datetime		Eintrag optional		Calculated Take Off Time
destination	datetime	4	Eintrag optional		Zielflughafen

Tabelle 5: Tabelle tdt_departure

Das Löschen eines Fluges in der Tabelle tdt_flight hat zur Folge, dass auch der entsprechende Tabelleneintrag in tdt_departure oder tdt_arrival gelöscht wird.

4.3 Tabelle tadd_aircraft_type

Die Information, um welchen Flugzeugtyp es bei einem Flug handelt, steht im Flugplan. Zusätzliche Angaben und Details zum jeweiligen Flugzeugtyp können für den Disponenten von Bedeutung sein und sind in der Tabelle tadd_aircraft_type hinterlegt.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_actype	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für den Flugzeugtyp (GUID)
type	varchar	45	Eintrag optional		Vierstelliger Code für den Typ des Flugzeugs (Bsp. A345)
flugzeug_id	varchar	20	Eintrag optional		Der in der Hamburger Flughafendatenbank FB2000 verwendeter Identifier für das Flugzeug
lfz_typ_id	varchar	20	Eintrag optional		Der in der Hamburger Flughafendatenbank FB2000 verwendeter Identifier für den Flugzeugtyp
wvc	varchar	5	Eintrag optional		Wirbelschleppenkategorie heavy = MTOW >= 136t medium = MTOW > 7t < 136t light = MTOW Bis 7t
average_seats	int	4	Eintrag optional		Anzahl Sitzplätze
description	varchar	70	Eintrag optional		Hersteller und detailliere Beschreibung für den type des Flugzeugs

Tabelle 6: Tabelle tadd_aircraft_type

Zu jedem Flugzeugtyp gibt es genau einen Tabelleneintrag mit guid_actype als Primärschlüssel. Alle weiteren Felder sind Detailinformationen zum jeweiligen Flugzeugtyp. Die tadd_aircraft_type ist die Parent-Tabelle, d.h. es muss gewährleistet sein, dass jeder Eintrag in dem Verknüpfungsfeld guid_ac_type der tdt_flight Tabelle auch als Eintrag in tadd_aircraft_type Tabelle existiert. Beide Tabellen stehen in einer 1:n-Beziehung zueinander, da jeder Flug nur ein Flugzeugtyp sein kann aber mehrere Flüge vom gleichen Flugzeugtyp sein können. Das Löschen eines Flugzeugtyps in der Tabelle tadd_aircraft_type kann nur dann erfolgen, wenn in der Tabelle tdt_flight kein Datensatz in dem Feld guid_ac_type den zu löschen guid_actype Tabelleneintrag enthält.

4.4 Tabelle tadd_position

Für die Ressourcenzuordnung von Gepäckfahrern zu Aufträgen der Gepäcktransportabwicklung ist es für den Disponenten von Bedeutung, die Standplätze der Fahrzeuge und Flugzeuge zu kennen. Die Tabelle tadd_position enthält alle von der FB 2000 kommenden Standplätze des Flughafens. Existieren zu den einzelnen Standplätzen zusätzliche Angaben und Details, so werden diese dem Disponenten über die Tabelle tadd_position zur Verfügung gestellt.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_position	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für Position (GUID)
pos_name	varchar	5	Eintrag optional		Name der Flughafenstandplätze
pos_nr	varchar	20	Eintrag optional		Nummer der Flughafenstandplätze
pos_description	varchar	45	Eintrag optional		Zum Namen der Flughafenstandplätze zusätzliche Detailinformationen
is_aircraft_pos	int	1	Eintrag optional		0 = Standplatz für Flugzeuge, 1 = sonst
is_pb_pos	int	1	Eintrag optional		0 = Standplatz für Pushback, 1 = sonst
is_bus_pos	int	1	Eintrag optional		0 = Standplatz für Busse, 1 = sonstiges
apron_number	varchar	5	Eintrag optional		Apronnummer, z.B. 1 = apron 1, 2 = apron 2, 4 = apron 4
av_pax_transport_dur	int	11	Eintrag optional		Zeitdauer des Passagiertransport
av_pb_transport_dur	int	11	Eintrag optional		Zeitdauer des Pushbackvorgangs

Tabelle 7: Tabelle tadd_position

Jeder Standplatz hat genau einen Datensatz in der Tabelle. Der Primärschlüssel ist guid_position, alle weiteren Felder sind Detailinformationen zum jeweiligen Standplatz. Die tadd_position ist die Parent-Tabelle, d.h. jedem Eintrag in dem Verknüpfungsfeld guid_actual_position und in dem Verknüpfungsfeld guid_planned_position der Child-Tabelle tdt_flight muss einem bestimmten Eintrag in tadd_position Tabelle zugeordnet werden.

Beide Tabellen stehen in einer 1:n-Beziehung zueinander, da jeder Flug nur jeweils einen aktuellen oder geplanten Standplatz haben kann, aber mehrere Flüge die gleichen aktuellen oder geplanten Standplätze haben können.

Das Löschen einer Position in der Tabelle tadd_position kann nur dann erfolgen, wenn in der Tabelle tdt_flight kein Datensatz in dem Feld guid_actual_position und guid_planned_position den zu löschenenden guid_position Tabelleneintrag enthält.

4.5 Tabelle tadd_luggage_belt

Die Tabelle tadd_luggage_belt enthält die zusätzlichen Angaben zu den einzelnen Gepäckbändern am Flughafen, die im Flugplan stehen können:

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_luggage_belt	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für das Gepäckband (GUID)
Belt_identifier	varchar	5	Eintrag optional		Gepäckbandnummer

Tabelle 8: Tabelle tadd_luggage_belt

Zu jedem Gepäckband gibt es genau ein Tabelleneintrag. Der Primärschlüssel ist guid_luggage_belt. Die tadd_luggage_belt ist die Parent-Tabelle, d.h der Fremdschlüssel in dem Verknüpfungsfeld guid_luggage_belt_to der Child-Tabelle tdt_flight kann nur solche Werte akzeptieren, die als Primärschlüssel in der tadd_luggage_belt Tabelle existieren.

Beide Tabellen stehen in einer 1:n-Beziehung zueinander, da zu jedem Flug für die Gepäcktransportabwicklung immer genau ein Gepäckband festgelegt ist, aber mehreren Flügen ein und dasselbe Gepäckband zugeordnet werden kann.

Das Löschen eines Gepäckbandes in der Tabelle tadd_luggage_belt kann nur dann erfolgen, wenn in der Tabelle tdt_flight kein Datensatz in dem Feld guid_luggage_belt_from und in dem Feld guid_luggage_belt_to den zu löschenenden guid_luggage_belt Tabelleneintrag enthält.

4.6 Tabelle tlnk_turnaround

Die Tabelle enthält eine Verknüpfung von zwei Flügen, die vom selben Luftfahrzeug durchgeführt werden. Hier ist z.B. hinterlegt, wie und wann ein Arrival wieder als Departure den Flughafen verlässt. Die Information darüber, ob ein Flug einen turnaround Flug hat oder nicht, ist für den Disponenten von großer Bedeutung, denn es gibt ihm die Möglichkeit,

- einen Überblick über die zur Abfertigung zur Verfügung stehende Zeit zu bekommen und
- ein und demselben Fahrer gleich zwei Aufträge zuzuweisen.
 - Das Gepäck für den departure vom Gepäckband abzuholen und zum entsprechenden Flugzeug hinzufahren
 - Vom gleichen Flugzeug das ausgeladene Gepäck sofort mitzunehmen und zum entsprechenden Gepäckband hinzufahren.

Die beiden Tabellen tdt_flight und tlnk-turnaround stehen in einer m:n-Beziehung zueinander, der Primärschlüssel ist guid_turnraound. In einer m:n Beziehung können jedem Datensatz in der tdt_flight Tabelle mehrere passende Datensätze in der tlnk_turnaround Tabelle zugeordnet sein und umgekehrt: Ein Flug kann 2 turnarounds haben, einmal als arrival und einmal als departure, ein turnaround wiederum hat 2 Flüge.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_turnaround	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für turnaround (GUID)
guid_arrival	varchar	40	Fremdschlüssel, Eintrag optional	Verweis auf Tabelle tdt_flight (guid_flight)	Identifier für Flug (GUID)
guid_departure	varchar	40	Fremdschlüssel, Eintrag optional	Verweis auf Tabelle tdt_flight (guid_flight)	Identifier für Flug (GUID)

Tabelle 9: Tabelle tlnk_turnaround

Das Löschen eines Fluges in der Tabelle tdt_flight kann nur dann erfolgen, wenn die Tabelle tlnk-turnaround kein guid_arrival Attribut und kein guid_departure Attribut im zu löschenen guid_flight Tabelleneintrag enthält.

4.7 Tabelle tadd_pax_flight

Das Management HMI soll dem Gepäcktransport Disponenten die Möglichkeit geben, eine Ressourcenzuordnung von Gepäckfahrern zu Fahraufträgen der Gepäcktransportabwicklung vorzunehmen. Für die Erstellung von Fahraufträgen müssen zwei Kriterien erfüllt sein:

- Es muss sich um einen Passagierflug handeln
- Im Flugplan muss die geplante Position bekannt sein

Ob der Flug ein Passagierflug ist oder nicht, ist in der Tabelle tadd_pax_flights hinterlegt.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_flight	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für Gepäcktransportauftrag (GUID)
passengers	int	11	Eintrag optional		Anzahl der Passagiere auf dem Flug

Tabelle 10: Tabelle tadd_pax_flights

Die beiden Tabellen tdt_flight und tadd_pax_flights stehen in einer 1:1-Beziehung zueinander, denn:

- jeder Flug ist entweder ein Passagierflug oder nicht
- jeder Passagierflug ist ein Flug.

4.8 Tabelle tdt_sita_messages

Zu einem Flug verfasste SITA-Meldungen sollten dem Disponenten jederzeit zur Verfügung stehen, denn sie geben Aufschluss über die Gepäckart, Gepäckstückzahlen etc. Es existieren nicht zu jedem Flug auch SITA- Meldungen. Doch alle vorhandenen SITA- Meldungen zu einem Flug werden in der tdt_sita_messages Tabelle gespeichert.

Beide Tabellen stehen in einer 1:n-Beziehung zueinander, da jede verfasste SITA-Meldung immer nur einem Flug zugeordnet werden kann, aber zu einem Flug kann es mehrere SITA-Meldungen geben.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_sita_messages	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für eine Sita-Meldung (GUID)
guid_flight	varchar	40	Fremdschlüssel, Eintrag Pflicht	Verweis auf Tabelle tdt_flight (guid_flight)	Identifier für den zugehörigen Flug (GUID)
sita_message_text	varchar	500	Eintrag optional		Der dazugehörige Inhalt der Sita-Meldung

Tabelle 11: Tabelle tdt_sita_messages

4.9 Tabelle tdt_task_luggage

Zu einem Flug kann nur ein Gepäcktransportauftrag erteilt werden, zur vollständigen Abwicklung eines Gepäcktransportauftrages kann es jedoch mehrere Fahraufträgen geben. Gepäcktransportaufträge werden in der Datenbank in der Tabelle tdt_task_luggage abgespeichert.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_task_luggage	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für Gepäcktransportauftrag (GUID)
guid_flight	varchar	40	Fremdschlüssel, Eintrag Pflicht	Verweis auf Tabelle tdt_flight (guid_flight)	Identifier für den zugehörigen Flug (GUID)
timestamp_requested	datetime		Eintrag Pflicht		Zeit, wann ein Fahrauftrag generiert wurde
timestamp_done	datetime		Eintrag optional		Zeit, wann ein Gepäcktransportauftrag beendet wurde
guid_pos_start	varchar	40	Eintrag Pflicht		GUID für die Position., von der Gepäck abgeholt werden soll
guid_pos_destination	varchar	40	Eintrag optional		GUID für die Position, zu der das Gepäck hingebracht werden soll
task_status	varchar	50	Eintrag Pflicht		open = Gepäcktransportauftrag ist noch in Arbeit und nicht beendet, done = Gepäcktransportauftrag ist beendet

Tabelle 12: Tabelle tdt_task_luggage

Das Feld guid_task_luggage ist der Primärschlüssel, jeder erzeugte Gepäcktransportauftrag ist eindeutig. Das Feld guid_flight ist der Fremdschlüssel und verweist auf die Parent-Tabelle tdt_flight, somit ist die Zuordnung des Auftrags zum Flug gewährleistet.

Das Löschen eines Fluges in der Tabelle tdt_flight hat zur Folge, dass auch der zugehörige Gepäcktransportauftrag in der tdt_task_luggage Tabelle automatisch mitgelöscht wird.

Die tdt_task_luggage Tabelle wird jeweils mit dem Feld guid_pos_start und mit dem Feld guid_pos_destination der Tabelle tadd_position verknüpft. Diese Tabelle ist eine zu der tdt_task_luggage Tabelle ergänzenden Tabelle. Tabelle 7 stellt den Aufbau dieser Nebentabelle dar.

4.10 Tabelle tdt_task_part_luggage_pax

Für die vollständige Abwicklung eines Gepäcktransportauftrags kann es mehrere Fahraufträge geben, jeder Fahrauftrag wird in der Datenbank in der Tabelle tdt_task_part_luggage_pax hinterlegt.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_task_part_luggage_pax	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für Fahraufträge (GUID)
guid_task_luggage	varchar	40	Fremdschlüssel, Eintrag Pflicht	Verweis auf Tabelle tdt_task_luggage (guid_task_luggage)	Identifier für den zugehörigen Gepäcktransportauftrag (GUID)
guid_vehicle_recommended	varchar	40	Eintrag optional		Identifier für das einem Fahrauftrag empfohlene Fahrzeug (GUID)
guid_vehicle_assigned	varchar	40	Eintrag optional		Identifier für das einem Fahrauftrag endgültig zugewiesene Fahrzeug (GUID)
arr_or_dep	varchar	3	Eintrag optional		Identifier, ob es sich bei dem Flug um ein arrival oder departure handelt.
timestamp_assigned	datetime		Eintrag optional		Zeit der Fahrzeugzuweisung vom Disponenten
timestamp_accepted	datetime		Eintrag optional		Zeit der Fahrzeugzuweisungsannahme vom Fahrer
timestamp_pick-up_luggage	datetime		Eintrag optional		Zeit, zu der der Fahrer das Gepäck abgeholt hat

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
timestamp_drop_luggage	datetime		Eintrag optional		Zeit, zu der der Fahrer das Gepäck abgegeben hat
timestamp_discontinued	datetime		Eintrag optional		Zeit, zu der der Fahrer den Fahrauftrag unterbrochen hat
timestamp_finished	datetime		Eintrag optional		Zeit, zu der der Fahrer den Fahrauftrag beendet hat
timestamp_displaced	datetime		Eintrag optional		Der vom Disponenten neu gesetzter Zeitpunkt der Fahrauftragsausführung
remarks	varchar	1000	Eintrag optional		Zum Fahrauftrag zusätzliche für den Groundhandling Disponent relevanten Informationen

Tabelle 13: Tabelle tdt_task_part_luggage_pax

Jeder erzeugter Fahrauftrag entspricht einem Datensatz in der tdt_task_part_luggage_pax Tabelle, die Eindeutigkeit des Datensatzes wird über den Primärschlüssel guid_task_part_luggage_pax ist der Primärschlüssel garantiiert. Das Feld guid_task_luggage ist der Fremdschlüssel und referenziert auf den Primärschlüssel der tdt_task_luggage Tabelle, damit ist die Zuordnung des Fahrauftrags zum Gepäcktransportauftrag gewährleistet.

Das Löschen eines Fluges in der Tabelle tdt_flight hat zur Folge, dass auch der zugehörige Gepäcktransportauftrag in der tdt_task_luggage Tabelle und alle zugehörigen Fahraufträge in der tdt_task_part_luggage_pax Tabelle automatisch mitgelöscht werden.

Die tdt_task_part_luggage_pax Tabelle wird jeweils:

- mit dem Feld guid_vehicle_recommended
- mit dem Feld guid_vehicle_assigned

der Tabelle tdt_vehicle verknüpft. Diese Tabelle ist eine ergänzende Tabelle zur tdt_task_part_luggage_pax, der Aufbau dieser Nebentabelle wird in Tabelle 16 dargestellt.

4.11 Tabelle tdt_task_part_luggage_abort

Ein Fahrauftrag kann während seiner Durchführung von dem betroffenen Fahrer abgebrochen werden.

Ein und derselbe Fahrauftrag kann mehrmals abgebrochen werden, aber den dazugehörigen Datenbankeintrag gibt es nur einmal. Ob ein Auftrag abgebrochen ist oder nicht erkennt das Programm an dem Feld timestamp_aborted:

- steht eine Zeit in dem Feld, so ist der Auftrag abgebrochen
- steht (NULL) in dem Feld, so ist der Auftrag wieder in Arbeit

Es liegt eine 1:1 Beziehung vor, somit ist der Primärschlüssel in tdt_task_part_luggage_pax als Feld in der Tabelle tdt_task_part_luggage_abort hinzuzufügen.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_task_part_luggage_pax	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für den zugehörigen task part (GUID)
reason	varchar	45	Eintrag optional		Grund, aus dem der Auftrag abgebrochen wurde.
timestamp_aborted	datetime		Eintrag optional	Ein Zeiteintrag bedeutet, dass der Auftrag abgebrochen ist. Steht dagegen Null drin, so ist der Auftrag wieder in Arbeit.	Zeit, zu der der Auftrag abgebrochen wurde.

Tabelle 14: Tabelle tdt_task_part_luggage_abort

Das Löschen eines Fahrauftrages in der Tabelle tdt_task_part_luggage_pax hat zur Folge, dass auch der entsprechende Tabelleneintrag in tdt_task_part_luggage_abort gelöscht wird.

4.12 Tabelle tdt_task_part_luggage_messages

Alle Meldungen, die für einen Fahrauftrages vorhandenen sind, sollten für den Disponenten jederzeit zur Verfügung stehen. Diese Meldungen werden in der tadd_task_part_luggage_messages Tabelle gespeichert.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_task_part_luggage_message	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für eine Meldung (GUID)
guid_task_part_luggage_pax	varchar	40	Fremdschlüssel, Eintrag Pflicht	Verweis auf Tabelle tdt_task_part_luggae_pax (guid_task_part_luggage_pax)	Identifier für den zugehörigen Fahrauftrag (GUID)
message_text	varchar	500	Eintrag optional		Der dazugehörige Inhalt der Meldung

Tabelle 15: Tabelle tdt_task_part_luggage_messages

Beide Tabellen stehen in einer 1:n-Beziehung zueinander, da jede verfasste Nachricht immer an einen Fahrauftrag gerichtet ist, aber zu einem Fahrauftrag kann es mehrere Meldungen geben.

4.13 Tabelle tdt_vehicle

Das Management HMI stellt alle auf dem Flughafen Hamburg zur Verfügung stehenden Gepäckfahrzeuge tabellarisch dar. Die zentrale Funktionalität der Fahrzeugtabelle besteht neben der Sichtungsmöglichkeit aktiver und inaktiver Fahrzeuge darin, Fahraufträge einzelnen Fahrern zuzuordnen. Die dargestellten Fahrzeuge werden aus der Datenbank ausgelesen, tdt_vehicle ist die Haupttabelle für die Fahrzeugdaten.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_vehicle	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für Fahrzeuge (GUID)
vehicle_number	varchar	10	Eintrag Pflicht		Fahrzeugnummer
vehicle_name	varchar	10	Eintrag Pflicht		Fahrzeugname
guid_currentPosition	varchar	40	Fremdschlüssel, Eintrag optional	Verweis auf Tabelle tadd_position (guid_position)	Identifier für die aktuelle Position des Fahrzeugs (GUID)

Tabelle 16: Tabelle tdt_vehicle

Für jedes Fahrzeug gibt es genau einen Datensatz in der Tabelle. Der Primärschlüssel ist guid_vehicle. Alle weiteren Felder sind Detailinformationen zum jeweiligen Fahrzeug. Die tadd_vehicle ist die Parent-Tabelle, d.h. jeder Eintrag in dem Verknüpfungsfeld guid_vehicle_recommended und in dem Verknüpfungsfeld guid_vehicle_assigned der Child-Tabelle tdt_task_part_luggage_pax muss einem bestimmten Eintrag in tdt_vehicle Tabelle zugeordnet werden.

Beide Tabellen stehen in einer 1:n-Beziehung zueinander, da jeder Fahrauftrag nur von einem Fahrzeug ausgeführt werden kann, aber ein Fahrzeug kann verschiedene Fahraufträge erledigen.

Das Löschen eines Fahrzeuges in der Tabelle tdt_vehicle kann nur dann erfolgen, wenn in der Tabelle tdt_task_part_luggage_pax kein Datensatz in dem Feld guid_vehicle_recommended und guid_vehicle_assigned den zu löschenenden guid_vehicle Tabelleneintrag enthält.

Durch die Spezifizierung, des Fahrzeugstatus und des dazugehörigen Fahrers, wird der tdt_vehicle Tabelle wird über die Tabelle tdt_luggagecar zusätzliche Informationen zugewiesen. Tabelle 17 beschreibt den Aufbau dieser Tabelle.

4.14 Tabelle tdt_luggagecar

Mit dem Management HMI wird in der Fahrzeugtabelle:

- der aktuelle Status der Gepäckfahrzeuge auf dem Vorfeld
- die, den Gepäckfahrzeugen aktuell zugeordneten Fahrer

dargestellt. Diese zusätzlichen Fahrzeuginformationen werden in der tdt_luggagecar Tabelle abgespeichert. Für jeden Datensatz in der Tabelle tdt_luggagecar gibt es nur einen entsprechenden Datensatz der tdt_vehicle Tabelle. Die tdt_luggagecar Tabelle ist eine Untermenge der tdt_vehicle Tabelle, denn wenn keine Werte für das Feld current_status und guid_current_employee vorhanden sind, soll es dementsprechend auch keinen Datensatz in der Tabelle tdt_luggagecar geben. Jeder Datensatz der tdt_vehicle ist nur mit maximal einem Datensatz in der tdt_luggagecar Tabelle verknüpft, die Eindeutigkeit der Verknüpfung wird über den Primärschlüssel guid_vehicle gewährleistet. Beide Tabellen stehen in einer 1:1 Beziehung zueinander.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_vehicle	varchar	40	Fremdschlüssel, Eintrag Pflicht	Verweis auf Tabelle tdt_vehicle (guid_vehicle)	Identifier für ein Fahrzeug (GUID)
current_status	varchar	30	Eintrag optional		Free = freies Fahrzeug, Proposed = vorgeschlagenes Fahrzeug etc.
guid_current_employee	varchar	40	Fremdschlüssel, Eintrag optional	Verweis auf Tabelle tdt_employee (guid_employee)	Der dazugehörige Fahrer (GUID)

Tabelle 17: Tabelle tdt_luggagecar

4.15 Tabelle tdt_employee

Ein Fahrzeug kann im Laufe eines Tages von verschiedenen Fahrern gefahren werden. Diese Verknüpfung erfolgt mit der Tabelle tdt_employee. Beide Tabellen stehen in einer 1:n-Beziehung zueinander, da ein Fahrer zu einem Zeitpunkt immer nur ein Fahrzeug fahren kann.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
guid_employee	varchar	40	Primärschlüssel, Eintrag Pflicht		Identifier für ein Fahrer (GUID)
personal_number	varchar	10	Eintrag optional		Personalnummer
surename	varchar	45	Eintrag optional		Nachname
forename	varchar	45	Eintrag optional		Vorname

Tabelle 18: Tabelle tdt_employee

4.16 Tabelle ttmp_updates_dlr Tabelle / ttmp_updates_uhh

Die beiden Tabellen ttmp_updates_dlr und ttmp_updates_uhh haben untereinander und zu den anderen Tabellen keine Beziehung. Sie dienen dem Austausch der Auftrags- und Positionsdaten mit dem Optimierer und dem SituationServer. Beide Tabellen haben den gleichen Aufbau.

Attribut	Datentyp	Länge	Eigenschaft	Erläuterungen	Beschreibung
id	bigint	20	Primärschlüssel, Eintrag Pflicht		Identifier für die update (GUID)
tbl_update	varchar	40	Eintrag optional		Name der Tabelle, wo ein Eintrag verändert wurde
guid_update	varchar	40	Eintrag optional		Identifier, welche Zeile der Eintrag verändert wurde (GUID)
insert_time	datetime		Eintrag optional		Zeit, wann der Eintrag in der ttmp_updates_dlr erfolgt ist
insert_reason	varchar	40	Eintrag optional		Art der Veränderung, definiert sind: insert, update, delete
abgearbeitet	tinyint	1	Eintrag optional		„0 = nicht abgearbeitet“, der Eintrag „1 = abgearbeitet“

Tabelle 19: Tabelle ttmp_updates_dlr / ttmp_updates_uhh

5 Anhang I

5.1 Abkürzungsverzeichnis / Glossar

CTOT	Calculated TakeOff Time
DLR	Deutsches Zentrum für Luft- und Raumfahrt
EIBT	Estimated Inblock Time
EOBT	Estimated Offblock Time
FB2000	Flughafendatenbank
FHG	Flughafen Hamburg GmbH
HMI	Human-Machine Interface
Mode-S-Transponder	Gerät im Flugzeug und speziellen fahrzeug zur Übertragung von Position und Identifikation
SIBT	Scheduled Inblock Time
SOBT	Scheduled Offblock Time
UHH	Universität Hamburg
WFF	“Wettbewerbsfähiger Flughafen”; Nachfolgeprojekt im Rahmen von Lufo IV der Bundesministeriums für Wirtschaft und Technologie (BMWI)
SQL	Structured Query Language
GUID	Globally Unique Identifier
SITA	Société Internationale de Télécommunications Aéronautiques

5.2 Abbildungsverzeichnis

Abbildung 1: Darstellung des Datenbankmodells.....	7
Abbildung 2: Primärschlüssel/Fremdschlüssel in einer Tabelle	9
Abbildung 3: Austauschtabellen	9

5.3 Tabellenverzeichnis

Tabelle 1: Verknüpfungen zwischen den einzelnen Tabellen.....	8
Tabelle 2: Tabellentypen.....	11
Tabelle 3: Tabelle tdt_flight.....	13
Tabelle 4: Tabelle tdt_arrival	14
Tabelle 5: Tabelle tdt_departure	14
Tabelle 6: Tabelle tadd_aircraft_type	15
Tabelle 7: Tabelle tadd_position	16
Tabelle 8: Tabelle tadd_luggage_belt	17
Tabelle 9: Tabelle tlnk_turnaround	18
Tabelle 10: Tabelle tadd_pax_flights.....	19

Tabelle 11: Tabelle tdt_sita_messages	20
Tabelle 12: Tabelle tdt_task_luggage	20
Tabelle 13: Tabelle tdt_task_part_luggage_pax.....	22
Tabelle 14: Tabelle tdt_task_part_luggage_abort.....	23
Tabelle 15: Tabelle tdt_task_part_luggage_messages	24
Tabelle 16: Tabelle tdt_vehicle	25
Tabelle 17: Tabelle tdt_luggagecar	26
Tabelle 18: Tabelle tdt_employee	27
Tabelle 19: Tabelle tttmp_updates_dlr / tttmp_updates_uhh	27