



Fabrika Güvenlik Takip Sistemi

Yazılım Mühendisliği Ana Bilim Dalı
Yüksek Lisans Bitirme Projesi

İpeksu KESKİN

Danışman: Doç. Dr. Aytuğ ONAN

Haziran 2023

İzmir Kâtip Çelebi Üniversitesi Fen Bilimleri Enstitüsü öğrencisi **İpeksu KESKİN** tarafından hazırlanan **Fabrika Güvenlik Takip Sistemi** başlıklı bu çalışma tarafımızca okunmuş olup, yapılan savunma sınavı sonucunda kapsam ve nitelik açısından başarılı bulunarak jürimiz tarafından YÜKSEK LİSANS olarak kabul edilmiştir.

ONAYLAYANLAR:

Proje Danışmanı:

Doç. Dr. Aytuğ ONAN
İzmir Kâtip Çelebi Üniversitesi

Yazarlık Beyanı

Ben, **İpeksu KESKİN**, başlığı **Fabrika Güvenlik Takip Sistemi** olan bu projemin ve projemin içinde sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu tezin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Tezin bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Kayda değer yardım aldığım bütün kaynaklara teşekkür ettim.
- Tezde başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

Tarih: 11.06.2023

Fabrika Güvenlik Takip Sistemi

ÖZ

Otomotiv parçası üreten bir fabrikada güvenlik ile ithalat ekibi arasında bir bağlantı olmamaktadır. Bu sebeple fabrikaya giren tırlardan ithalat ekibinin bilgisinin olmamasıyla yönetilmesi zor bir haldedir. Hem güvenlik hem de ithalat birimlerinin tuttıkları verileri kullanarak yeni bir database oluşturulmuştur. Burada her iki veride ortak olarak tır plakası vardır ve plakadan yola çıkılarak her günü içeren bir veri seti oluşturulmuştur. Böylelikle her iki birim için olması gereken en net en işe yarar bilgilerin veri seti MySQL ile oluşturulmuştur.

Anahtar Sözcükler: MySQL, İthalat, Güvenlik, Veri Seti, Fabrika, Fabrika Güvenlik Birimi, Fabrika İthalat Birimi, Tır Girişi , Plaka Numarası

Factory Security Tracking System

Abstract

In a factory producing automotive parts, there is no connection between the security and the import team. Therefore, it is difficult to manage with the lack of knowledge of the import team from the trucks entering the factory. A new database has been created using the data kept by both the security and import teams. Here, the truck plate is common in both data and a data set containing each day has been created based on the plate. These are the data set of the clearest and most useful information that should be for both teams.

Keywords: MySQL, Import, Security, Data Set, Factory, Factory Security Team, Factory Import Team, Truck Entry, Plate Number

Teşekkür

Bitirme projesi çalışmasına katkılarından dolayı İzmir Katip Çelebi Üniversitesi Yazılım Mühendisliği Tezsiz Yüksek Lisans bölümünde dersini aldığım tüm öğretmenlere verdikleri emek ve anlayışları için teşekkür ederim.

İçindekiler

Yazarlık Beyanı	ii
Öz	iii
Abstract	iv
Teşekkür	v
Şekiller Listesi.....	vii
Kısaltmalar Listesi	viii
1 Bölüm 1- Giriş	1
2 Bölüm 2 – SQL Tanımı , SQL Veri Tabanı	3
2.1 SQL Tanımı	3
2.2 SQL Sorguları	4
2.3 SQL Veri Tipleri	5
2.4 SQL Veri Tabanı.....	6
2.4.1 SQL Veri Tabanı Faydaları	7
2.5 SQL Veri Tabanının Fabrika Güvenlik Sisteminde Kullanılması	9
3 Bölüm 3 – Fabrika Güvenlik Takip Sistemi	10
3.1 Güvenlik Veri Seti	10
3.2 İthalat Veri Seti	13
3.3 MySQL Güvenlik Veri Sisteminin Oluşması	13
3.3.1 İthalat (Import) Tablosunun Oluşturulması	14
3.3.2 İthalat (Import) Tablosuna Veri Girişi.....	15
3.3.3 Güvenlik (Security) Tablosunun Oluşturulması.....	17
3.3.4 Güvenlik (Security) Tablosuna Veri Girişi	19
3.2.5 Genel (Generall) Tablosunun Oluşturulması.....	21
Kaynaklar	24

Şekiller Listesi

Şekil 1	Otomotiv parçası üreten bir firmaya ait güvenlik araç giriş çıkışını gösteren akış.....	11
Şekil 2	Otomotiv parçası üreten bir firmaya ait güvenlik araç giriş çıkışını gösteren akış (devamı)	12
Şekil 3	MySQL importdb kodu	13
Şekil 4	İthalat (Import) tablosunun MySQL kodu	15
Şekil 5	MySQL İthalat (Import) tablosunun içine veri girme	16
Şekil 6	MySQL İthalat (Import) tablosunun içine veri girme(devamı).....	16
Şekil 7	MySQL İthalat (Import) tablosunun çıktısı.....	17
Şekil 8	Güvenlik (Security) tablosunun MySQL kodu	18
Şekil 9	MySQL Güvenlik (Security) tablosunun içine veri girme	19
Şekil 10	MySQL Güvenlik (Security) tablosunun içine veri girme(devamı).....	20
Şekil 11	MySQL Güvenlik (Security) tablosunun çıktısı.....	20
Şekil 12	MySQL importdb ye bağlı tablolar	21
Şekil 13	Genel (Generall) tablosunun MySQL kodu	21
Şekil 14	MySQL Genel (Generall) tablosunun çıktısı	22

Kısaltmalar Listesi

SQL	Yapısal Sorgulama Dili
RDBMS	İlişkisel Veritabanı Yönetim Sistemleri
İKÇÜ	İzmir Kâtip Çelebi Üniversitesi
DB	Database (Veri Seti)

Bölüm 1

Giriş

Bu proje otomotiv parçası üreten bir fabrikanın güvenlik biriminde tır girişlerini manuel olarak deftere kaydetmesi , belirli bir sistematığı olmaması ve geriye yönelik veri takibi olmadığı için MySQL de veri tabanı oluşturulmasını amaçlamaktadır.Bu veri tabanı oluşturulurken tır giriş çıkışlarını ithalat biriminin excel verilerinden yararlanılmıştır.Burada amaç ithalat birimi ile güvenlik birimi arasında doğru bağlantı kurmak ve araç giriş çıkışlarından ilgili birimlerin haberinin olmasının sağlamak.Yapılacak olan veri tabanı ile ithalat birimi kendi belirlediği tarih ve saatte tırın giriş çıkış yapıp yapmadığını kontrol edecek ve süreçler daha hızlı ilerleyecektir.

Günümüzde, fabrikalar ve endüstriyel tesisler, çalışanların güvenliği ve işyeri güvenliği konularında önemli bir öncelik haline gelmiştir. Fabrikalarda gerçekleşebilecek kazaların potansiyel riskleri ve olumsuz sonuçları göz önüne alındığında, etkili bir güvenlik takip sisteminin kullanılması hayati önem taşımaktadır. Bu tez, fabrika güvenlik takip sisteminin tasarımı ve uygulanması üzerine odaklanmayı amaçlamaktadır.

Bu çalışmada, fabrika güvenlik takip sisteminin önemi ve gerekliliği vurgulanmaktadır. Geleneksel güvenlik yöntemleri genellikle yetersiz kalırken, otomatik ve entegre bir güvenlik takip sistemi, fabrikalarda güvenliği artırmak ve kazaların önlenmesine yardımcı olmak için etkili bir çözüm sunmaktadır. Bu projede, fabrika güvenlik takip sisteminin bileşenleri, işleyişi ve sağladığı avantajlar ayrıntılı olarak incelenmiştir.

Ayrıca, fabrika güvenlik takip sistemi için kullanılan teknolojiler ve algoritmalar üzerinde durulmuştur.Veritabanı analitiği ve yapay zeka gibi teknolojiler, güvenlik takip sisteminin etkinliğini artırmak için kullanılan temel araçlardır. Bu tezde,iki veri tabanı

arasında ilişki kurup tek bir veri tabanıla herkesin oluşmak istediği nihai sonucu gösteren bir veri tabanı hazırlamaktır.

Son olarak, projenin amacı, gerçek bir fabrika problemidir ve buna çözüm olarak tasarlanmıştır. Güvenlik biriminin tır giriş çıkışlarından haberdar olması için takip sistemi gerekmektedir bu sebeple veri tabanı oluşturulması amaçlanmıştır. Bu çalışmada, bir fabrika ortamında güvenlik takip sistemi için bir prototip geliştirilerek, sistemin nasıl çalıştığı ve nasıl kullanıldığı gösterilmiştir.

Bu proje, fabrika güvenlik takip sistemi konusunda önemli bir katkı sağlamayı hedeflemektedir. Sonuç olarak, fabrikalarda güvenlik standartlarının artırılması ve çalışanların güvenliğinin sağlanması için otomatik ve entegre bir güvenlik takip sisteminin önemi vurgulanmaktadır.

Bölüm 2

SQL Tanımı , SQL Veri Tabanı

Bu bölümde, Fabrika Güvenlik Takip Sisteminde veritabanı oluşturulurken kullanılan SQL dili hakkında detaylı bilgi vermek ve faydalarından bahsedilmesi amaçlanmıştır.

2.1 SQL Tanımı

SQL (Structured Query Language), ilişkisel veritabanı yönetim sistemlerinde (RDBMS) kullanılan bir programlama dilidir. SQL, veritabanı oluşturma, veri ekleme, güncelleme, sorgulama, veri silme gibi işlemleri gerçekleştirmek için kullanılır.

SQL, veritabanı yönetim sistemleriyle (örneğin, MySQL, PostgreSQL, Oracle, Microsoft SQL Server) etkileşim kurmanın standart yoludur. Veritabanı oluşturma ve tablo tanımlama, veri ekleme, güncelleme, sorgulama, silme, sıralama, filtreleme gibi işlemler SQL sorgularıyla gerçekleştirilir.

SQL, Structured Query Language yani "Yapılandırılmış Sorgu Dili" veritabanlarında kullanılan bir prosedürdür. Çoğu veritabanı SQL komutları ile çalışır. SQL Sorguları ile veritabanlarına bilgiler eklenir, silinir ve güncellenir. Kısacası, SQL çoğu Veritabanlarında kullanılan komut dilidir. MySQL de bir veritabanıdır. Unix tabanlı sistemler için geliştirilen ancak zaman içinde Win, Mac gibi pek çok sistemde kullanılabilir hale gelen MySQL veritabanları arasındaki en hızlı dillerdendir. (Ekinci 2006).[1]

MySQL, çok hızlı ve sağlam bir ilişkisel veritabanı yönetim sistemidir. Bir veritabanı, her türlü veriyi saklamayı, aramayı ve elde etmeyi sağlar. MySQL sunucusu, verilere erişimi kontrol ederken çok sayıda kullanıcının aynı anda üzerinde çalışmasını ve hızlı erişimi sağlar. Ayrıca aynı anda sadece yetkili kullanıcıların erişim sahibi olmalarını güvence altına alır. Bu yüzden MySQL, çoklu kullanıcı, çoklu kanallı bir sunucudur. (Welling and Thomson, 2002)[2]

2.2 SQL Sorguları

SQL sorguları, tablolardaki verilere erişmek ve işlem yapmak için kullanılır. Sorgular, SELECT, INSERT, UPDATE, DELETE gibi komutlardan oluşur. Yaygın olarak kullanılan SQL sorgu türleri :

SELECT: Veritabanından veri çekmek için kullanılır.

"SELECT * FROM tablo_adi" (tablo_adi yerine ilgili tablo adı kullanılır).

INSERT: Veritabanına yeni veri eklemek için kullanılır.

"INSERT INTO tablo_adi (sütun1, sütun2) VALUES (değer1, değer2)".

UPDATE: Varolan verileri güncellemek için kullanılır.

"UPDATE tablo_adi SET sütun1 = yeni_değer WHERE koşul".

DELETE: Veritabanından veri silmek için kullanılır.

"DELETE FROM tablo_adi WHERE koşul".

CREATE DATABASE: Yeni bir veritabanı oluşturmak için kullanılır.

"CREATE DATABASE mydatabase"

DROP DATABASE: Var olan bir veritabanını silmek için kullanılır.

"DROP DATABASE mydatabase"

CREATE TABLE: Yeni bir tablo oluşturmak için kullanılır.

"CREATE DATABASE import"

DROP TABLE: Var olan bir tabloyu silmek için kullanılır.

"DROP TABLE customers"

ALTER TABLE: Var olan bir tabloyu değiştirmek için kullanılır.

"ALTER TABLE customers ADD COLUMN phone VARCHAR(20) "

JOIN: İki veya daha fazla tabloyu birleştirmek için kullanılır.

JOIN komutu birden fazla türle ifade edilebilir , JOIN türleri ;

INNER JOIN: İki tablonun ortak kayıtlarını birleştirir.

LEFT JOIN: Sol tablodaki kayıtların sağ tablodaki eşleşen kayıtları ile birleştirir. Eşleşme olmadığı durumlarda sol tablonun kayıtları görünür, sağ tablo kayıtları NULL değerlerle temsil edilir.

RIGHT JOIN: Sağ tablodaki kayıtların sol tablodaki eşleşen kayıtları ile birleştirir. Eşleşme olmadığı durumlarda sağ tablonun kayıtları görünür, sol tablo kayıtları NULL değerlerle temsil edilir.

FULL JOIN: Hem sol hem de sağ tablodaki tüm kayıtları birleştirir. Eşleşme olmadığı durumlarda NULL değerler kullanılır.

SQL, veritabanlarında veri yapılarını (tablolar, sütunlar, satırlar) tanımlamak ve bu verilerle etkileşimde bulunmak için güçlü bir araçtır. Aynı zamanda veritabanları arasında ilişkiler kurma, verileri sorgulama ve raporlama gibi ileri düzey işlemleri de destekler.

SQL, standart bir dil olup, birçok veritabanı yönetim sistemi tarafından desteklenir. Bununla birlikte, bazı farklılıklar olabilir ve her veritabanı yönetim sistemi bazında özel dil özellikleri de bulunabilir.

2.3 SQL Veri Tipleri

Veri tipleri, verilerin doğru şekilde depolanması, sorgulanması ve işlenmesi için önemlidir. SQL de kullanılan yaygın veri tipleri :

INTEGER: Tamsayı değerleri temsil etmek için kullanılır.

FLOAT: Küsuratlı sayıları temsil etmek için kullanılır.

DOUBLE: Çift hassasiyetli küsuratlı sayıları temsil etmek için kullanılır.

DECIMAL: Hassas ksratlı sayıları temsil etmek iin kullanılır.

CHAR: Sabit uzunluklu karakter dizilerini temsil etmek iin kullanılır.

VARCHAR: Deęişken uzunluklu karakter dizilerini temsil etmek iin kullanılır.

TEXT: Uzun metin verilerini temsil etmek iin kullanılır.

DATE: Yıl, ay ve gn bilgilerini temsil etmek iin kullanılır.

TIME: Saat, dakika ve saniye bilgilerini temsil etmek iin kullanılır.

DATETIME: Tarih ve saat bilgilerini temsil etmek iin kullanılır.

TIMESTAMP: Zaman damgası (timestamp) deęerlerini temsil etmek iin kullanılır.

BOOLEAN: Mantıksal (true/false) deęerleri temsil etmek iin kullanılır.

PRIMARY KEY: Tekil tanımlayıcı olarak kullanılan ve bir tablodaki benzersiz kayıtları temsil eden bir stn.

FOREIGN KEY: İlişkili tablolardaki kayıtları birbirine bağlamak iin kullanılan bir stn.

AUTO_INCREMENT: Bir stnn otomatik olarak artan deęerlerle doldurulmasını saęlayan bir zellik.

2.4 SQL Veritabanı

SQL veritabanı, ilişkisel veritabanı ynetim sistemleri (RDBMS) tarafından kullanılan bir tr veri depolama ve ynetim sistemi olarak tanımlanır. SQL (Structured Query Language) dili kullanılarak oluřturulan ve iřletilen veritabanlarıdır.

SQL veritabanları, verileri tablolara, stnlara ve satırlara blerek dzenler. Bu tablolarda veriler ilişkisel olarak temsil edilir ve veri btnlęn saęlamak iin

kısıtlamalar uygulanır. SQL veritabanları, verilere erişmek, veri eklemek, güncellemek, sorgulamak, silmek ve veri ilişkilerini yönetmek için SQL sorguları kullanılır.

SQL veritabanlarının bazı özellikleri şunlardır:

Tablo Yapısı: Veriler tablolar halinde saklanır. Her tablo, belirli bir veri türüne veya konuya ilişkin bilgileri içeren sütunlardan oluşur.

İlişkiler: Tablolar arasında ilişkiler kurulabilir. Bu, verilerin bir tablodan diğerine referanslarla bağlanmasını sağlar ve veri bütünlüğünü korur.

Kısıtlamalar: Veritabanında kısıtlamalar tanımlanabilir. Örneğin, benzersizlik kısıtlaması, bir sütundaki değerlerin benzersiz olmasını sağlar. İlişkisel kısıtlamalar, veri bütünlüğünü sağlamak için kullanılır.

Veri Güvenliği: SQL veritabanları, kullanıcıların verilere erişimini ve yetkilerini yönetme imkanı sağlar. Kimlik doğrulama, yetkilendirme ve erişim denetimi gibi güvenlik önlemleri uygulanabilir.

Veri Yedekleme: Veritabanları, verilerin düzenli olarak yedeklenmesini sağlar. Bu, veri kaybı durumunda verilerin kurtarılmasını ve iş sürekliliğini sağlar.

Performans Optimizasyonu: SQL veritabanları, veri sorgularını optimize etmek için endeksleme, sorgu optimizasyonu ve veritabanı indeksleri gibi teknikleri kullanır. Bu sayede sorgu performansı artırılabilir.

SQL veritabanları, geniş bir uygulama alanına sahiptir. İşletmeler, web siteleri, mobil uygulamalar, bankalar, e-ticaret platformları ve daha pek çok alanda veri depolama ve yönetim ihtiyaçlarını SQL veritabanlarıyla karşılarlar.

2.4.1 SQL Veritabanı Faydaları

SQL'de veritabanı oluşturmanın faydaları ve amacı şunlardır:

Veri Yönetimi: Veritabanı oluşturarak, verileri düzenli bir şekilde saklama ve yönetme imkanı elde edersiniz. Veritabanı, verilerin tablolar, sütunlar ve satırlar şeklinde

organize edilmesine olanak sağlar. Bu sayede verilere kolay erişim, sorgulama, güncelleme ve silme işlemleri gerçekleştirilebilir.

Veri Bütünlüğü: Veritabanı, verilerin bütünlüğünü koruma amacıyla kullanılır. Veritabanında tanımlanan kısıtlamalar (örneğin, benzersizlik kısıtlaması veya referans bütünlüğü) sayesinde, yanlış veya tutarsız verilerin veritabanına eklenmesi veya güncellenmesi engellenir. Bu, veri bütünlüğünün sağlanmasına yardımcı olur.

Veri Paylaşımı ve İşbirliği: Veritabanı, birden fazla kullanıcının aynı verilere erişmesine ve üzerinde çalışmasına olanak tanır. Bu, ekip çalışması ve veri paylaşımı için önemlidir. Farklı kullanıcılar, aynı veritabanına bağlanarak veri güncellemelerini, sorgulamaları ve raporlamayı gerçekleştirebilir.

Veri Güvenliği: Veritabanları, verilerin güvenli bir şekilde depolanmasını ve erişimin kontrol altında tutulmasını sağlar. Kullanıcılar ve roller aracılığıyla, belirli kullanıcıların hangi verilere erişebileceği ve hangi işlemleri gerçekleştirebileceği kontrol edilebilir. Veritabanı, yetkilendirme ve kimlik doğrulama gibi güvenlik önlemlerini destekler.

Performans ve Veri Analizi: Veritabanları, verilere hızlı ve etkili bir şekilde erişimi sağlar. Veritabanı yönetim sistemleri, verileri optimize ederek sorgu performansını artırır. Ayrıca, veri analitiği ve raporlama için SQL sorguları kullanılarak verilerin derinlemesine analizi yapılabilir.

Veri Yedekleme ve Kurtarma: Veritabanları, verilerin yedeklenmesini ve kurtarılmasını kolaylaştırır. Düzenli yedeklemeler sayesinde, veri kaybı durumunda verilerin geri alınması mümkün olur. Bu, iş sürekliliği ve veri güvenliği açısından önemlidir.

Veritabanı oluşturmanın amacı, verileri düzenli bir şekilde saklamak, erişmek, güncellemek, silmek ve analiz etmek için bir yapı sağlamaktır.

2.5 SQL Veritabanının Fabrika Güvenlik Sisteminde Kullanılması

Fabrika güvenlik sistemi genellikle farklı bileşenlerden oluşur ve birçok veriyi işler. SQL, bu verilerin etkin bir şekilde saklanması, yönetilmesi için kullanılabilir.

Fabrika güvenlik sistemi için kullanılacak bazı SQL kullanım örnekleri vardır bunlar :

Kullanıcı Yönetimi: Fabrika güvenlik sistemi genellikle farklı kullanıcılar ve erişim düzeyleri içerir. SQL, kullanıcı hesaplarının ve yetkilerinin saklanması için kullanılabilir.

Güvenlik Olaylarının Kaydedilmesi: Fabrika güvenlik sistemi, çeşitli güvenlik olaylarını (örneğin, giriş-çıkış kayıtları, alarm durumları, geçişler vb.) kaydetmek için SQL veritabanını kullanabilir. Bu olaylar tablolar ve sütunlar şeklinde yapılandırılabilir ve ilgili verilerin zaman damgasıyla kaydedilmesi sağlanabilir.

Erişim Kontrolü: SQL, fabrika güvenlik sisteminde erişim kontrollerini uygulamak için kullanılabilir.

Sorgulama ve Raporlama: SQL, fabrika güvenlik sistemindeki verileri sorgulamak ve raporlamak için kullanılabilir. Güvenlik olayları, geçiş kayıtları, alarm durumları gibi verilerin sorgulanması ve analiz edilmesi, SQL sorguları aracılığıyla gerçekleştirilebilir.

SQL, fabrika güvenlik sisteminin veri yönetimi, erişim kontrolü, olay kaydı ve raporlama gibi önemli işlevlerinde kullanılabilir. Verilerin düzenli bir şekilde saklanması, güncellenmesi ve analiz edilmesi için SQL kullanımı, sistemin etkinliğini ve güvenliğini artırabilir.

Bölüm 3

Fabrika Güvenlik Takip Sistemi

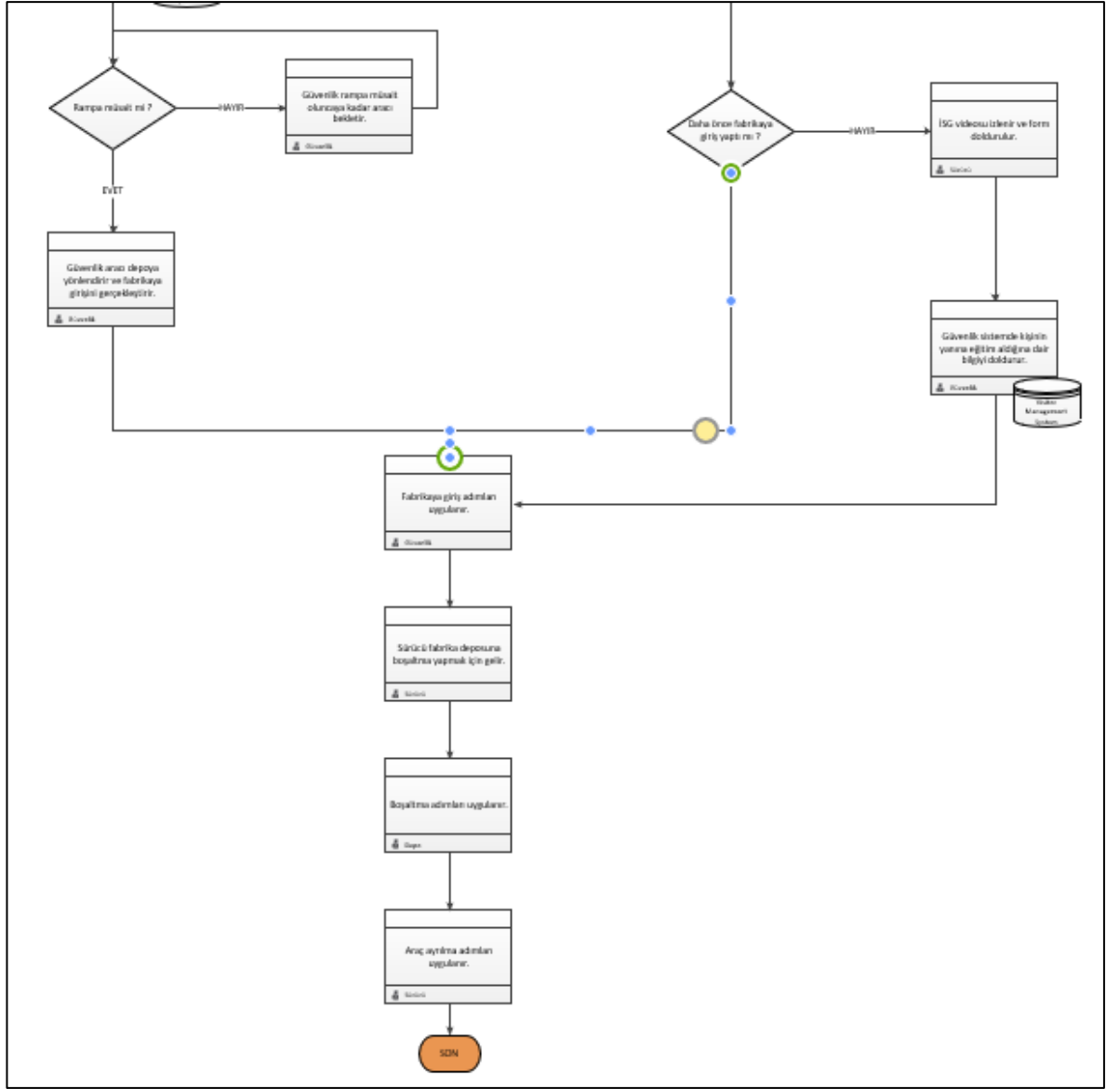
Otomotiv parçası üreten bir firmada güvenliğin tuttuğu verilerle ithalatın tuttuğu verileri veri tabanı oluşturma daha sonra iki veri arasında takip sistemi oluşturan bir veri tabanı sistemi yaratmak amaçlanmıştır.Böylelikle firma bu ortak veri tabanıyla hangi tırın saat kaçta burda olduğu bilgisini net şekilde görecektir.

3.1 Güvenlik Veri Seti

Güvenliğin tır giriş çıkışı da dahil olmak üzere fabrikaya giren tüm araçların giriş çıkışı bir excel takip sistemine göre çalışmaktadır.Bu exceldeki takip sistemine göre giriş yapan kişinin adı soyadı,giriş yapan tırın firma bilgileri, TC no, plaka numarası,giriş zamanı, çıkış zamanı,fabrikaya geliş nedeni,kiminle görüşeceği,iş sağlığı güvenliği eğitim videosu izletildi bilgisi vb. bilgileri kaydettiği bir excel dosyası vardır.

Güvenlik birimi bu veri setini doldururken gelen araçları görüşeceği kişilere bilgi vererek içeri almaktadır ancak tır girişlerinde kimseden onay almadan sadece depo ile rampanın boş olup olmadığı bilgisini öğrendikten sonra içeri almaktaydı.Bu sebeple tır giriş çıkışları yönetilemiyor hangi tırın ne zaman geldiği hangi zaman çıkış yaptığı fabrika içinde ve sanayi bölgesinde ne kadar beklediğiyle ilgili faturalar geldiğinde belli olmaktaydı.Bu sebeple ithalat biriminin hangi tırın ne zaman geldiğini güvenlik biriminden telefon ile iletişime geçerek öğrenebiliyordu.

Otomotiv parçası üreten bu firmaya ait güvenlik araç ve tır giriş çıkışlarını gösteren akışta ;



Şekil 2: Otomotiv parçası üreten bir firmaya ait güvenlik araç giriş çıkışını gösteren akış (devamı)

3.2 İthalat Veri Seti

İthalat biriminin tır giriş çıkışı da dahil olmak üzere fabrikaya giren tüm tırların girişi için bir excel takip sistemine göre çalışmaktadır. Bu exceldeki takip sistemine göre giriş yapan tırın referans numarası , tedarikçi adı, sevkiyatçı adı, plaka numarası , hangi yol tipi ile geldiği , gelen tırdaki paket sayısı, gelen paketin ağırlığı , ticari fatura numarası , geliş tarihi vb. bilgileri kaydettiği bir excel dosyası vardır.

Bu exceli her gün bilgi değişikçe ve takip edebilmek amacıyla tutmaktadırlar. Yaptıkları bu günlük plana göre giriş yapan araçları güvenliği arayarak ya da depo ile iletişim kurarak takip etmektedirler.

3.3 MySQL Güvenlik Veri Sisteminin Oluşması

MySQL ile veri setinin oluşturulabilmesi için ilk olarak create database komutu kullanılarak “importdb” isimli bir database yaratılmıştır.

```
1 • create database importdb;
```

Şekil 3 : MySQL importdb kodu

Importdb isimli database yaratıldıktan sonra güvenlik ve ithalat ekibine ait tabloları importdb veri tabanına bağlı olacaktır.

3.3.1 İthalat (Import) Tablosunun Oluşturulması

Importdb yaratıldıktan sonra “use” komutuyla yaratılan ithalat tablosunun importdb ye bağlanmaktadır.CREATE TABLE komutunu kullanarak IMPORT adlı tablo oluşmaktadır.IMPORT tablosunda ithalat ekibinin mevcutta var olan excellerinden veri setinde olması gereken sütunlar eklenmiştir.

REF_NO : referans numarasıdır ve primary key olarak tanımlanmıştır.Bu da referans numarasının benzersiz kayıt olduğunu göstermektedir.

FORWARDER_NAME : sevkiyatçının adıdır varchar(255) 255 uzunluklu karakter dizilimi komutunu kullanarak tanımlanmıştır.NOT NULL ifadesiyle bu verinin mutlaka dolu olması gerekmektedir.

ARRIVAL_TIME : tırın gelme tarihi olarak DATE komutuyla tanımlanmıştır. Veri seti için önemli bir veri olduğundan NOT NULL ifadesiyle bu verinin mutlaka dolu olması gerekmektedir.

CUP_NO : tırda gelen yükletin kap adedi olarak tanımlanmıştır.Küçük tamsayı değeri olduğu için TINYINT olarak tanımlanmıştır.

PLATE_NO : Plaka numarası TEXT olarak ve veri setinin en önemli verisi olduğu için NOT NULL olarak tanımlanmıştır.

WEIGHT : tırda gelen yüklerin ağırlığı olarak tanımlanmıştır.Küçük tamsayı değeri olduğu için TINYINT olarak kullanılmıştır.

```
1 • use importdb;
2 • CREATE TABLE IF NOT EXISTS IMPORT(
3   REF_NO INT AUTO_INCREMENT PRIMARY KEY ,
4   FORWARDER_NAME VARCHAR(255) NOT NULL ,
5   ARRIVAL_TIME DATE NOT NULL,
6   CUP_NO TINYINT ,
7   PLATE_NO TEXT NOT NULL ,
8   WEIGHT TINYINT);
```

Şekil 4 : İthalat (Import) tablosunun MySQL kodu

3.3.2 İthalat (Import) Tablosuna Veri Girişi

İthalat tablosu tanımlandıktan sonra içine verileri INSERT INTO komutunu kullanarak aktarılmıştır.

INSERT INTO IMPORT ifadesiyle import olarak tanımlanan tablonun içerisine veri ekleme komutu olarak tanımlanmıştır.Sonrasında import tablosuna ait oluşturulan sütunları parantez içinde sırayla belirtilmiştir.

Alt satıra geçilirken VALUES komutu ile tanımlanan sütunlara sırasıyla ' ' , kullanılarak istenilen veriler yazılır.Yazılan verilerden sonra uygun komutlara ve sıraya göre veriler tablonun içine aktarılmıştır.


```

1 • use importdb;
2 • CREATE TABLE IF NOT EXISTS IMPORT(
3     REF_NO INT AUTO_INCREMENT PRIMARY KEY ,
4     FORWARDER_NAME VARCHAR(255) NOT NULL ,
5     ARRIVAL_TIME DATE NOT NULL,
6     CUP_NO TINYINT ,
7     PLATE_NO TEXT NOT NULL ,
8     WEIGHT TINYINT);
9
10
11 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
12     VALUES ('128', 'EKOL', '2023-01-27', '7', '35İKL15', '12');
13 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
14     VALUES ('129', 'ALISAN', '2023-01-28', '7', '35AAC684', '13');
15 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
16     VALUES ('130', 'ANS', '2023-01-29', '7', '35AAN184', '14');
17 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
18     VALUES ('131', 'ARKAS', '2023-01-30', '9', '35AAU760', '15');
19 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
20     VALUES ('132', 'EKOL', '2023-01-31', '12', '35AAZ077', '19');
21 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
22     VALUES ('133', 'ATOM', '2023-02-03', '15', '35ABF18', '12');
23 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
24     VALUES ('134', 'ATOM LOJISTIK', '2023-02-09', '1', '35ABJ299', '25');

```

Şekil 5 : MySQL İthalat (Import) tablosunun içine veri girme

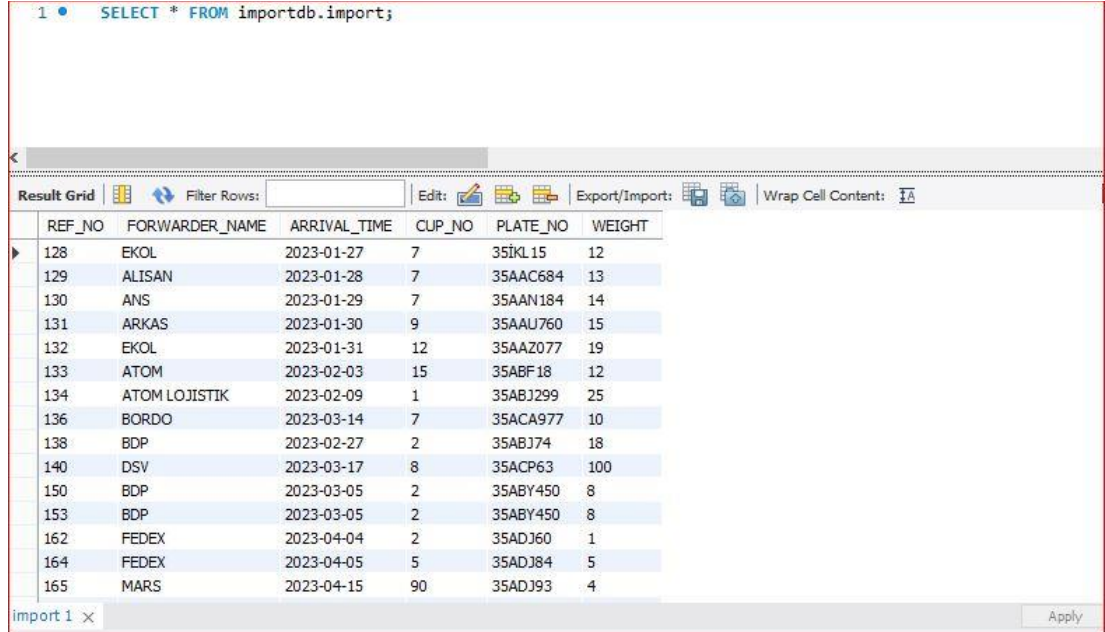
```

25 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
26     VALUES ('153', 'BDP', '2023-03-05', '2', '35ABY450', '8');
27 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
28     VALUES ('136', 'BORDO', '2023-03-14', '7', '35ACA977', '10');
29 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
30     VALUES ('140', 'DSV', '2023-03-17', '8', '35ACP63', '10');
31 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
32     VALUES ('145', 'DSV', '2023-03-27', '15', '35ACP64', '20');
33 • INSERT INTO IMPORT(REF_NO, FORWARDER_NAME, ARRIVAL_TIME, CUP_NO, PLATE_NO, WEIGHT)
34     VALUES ('146', 'EGE EKSPRES', '2023-03-30', '17', '35ADJ299', '25');

```

Şekil 6 : MySQL İthalat (Import) tablosunun içine veri girme (devamı)

SELECT*FROM importdb.import komutu ile importdb veri tabanının içinden import adlı tablodan tüm sütunları getir adlı komutu kullanarak verilerin tabloya işleyip işlemediği kontrol edilmiştir.



```
1 • SELECT * FROM importdb.import;
```

REF_NO	FORWARDER_NAME	ARRIVAL_TIME	CUP_NO	PLATE_NO	WEIGHT
128	EKOL	2023-01-27	7	35İKL15	12
129	ALISAN	2023-01-28	7	35AAC684	13
130	ANS	2023-01-29	7	35AAN184	14
131	ARKAS	2023-01-30	9	35AAU760	15
132	EKOL	2023-01-31	12	35AAZ077	19
133	ATOM	2023-02-03	15	35ABF18	12
134	ATOM LOJISTIK	2023-02-09	1	35ABJ299	25
136	BORDO	2023-03-14	7	35ACA977	10
138	BDP	2023-02-27	2	35ABJ74	18
140	DSV	2023-03-17	8	35ACP63	100
150	BDP	2023-03-05	2	35ABY450	8
153	BDP	2023-03-05	2	35ABY450	8
162	FEDEX	2023-04-04	2	35ADJ60	1
164	FEDEX	2023-04-05	5	35ADJ84	5
165	MARS	2023-04-15	90	35ADJ93	4

Şekil 7 : MySQL İthalat (Import) tablosunun çıktısı

3.3.3 Güvenlik (Security) Tablosunun Oluşturulması

Importdb yaratıldıktan sonra “use” komutuyla yaratılan güvenlik tablosunun importdb ye bağlanmaktadır.CREATE TABLE komutunu kullanarak SECURITY adlı tablo oluşmaktadır.SECURTIY tablosunda güvenlik ekibinin mevcutta var olan excellerinden veri setinde olması gereken sütunlar eklenmiştir.

DRIVER_NAME : tır sürücüsünün adıdır varchar(255) 255 uzunluklu karakter dizilimi komutunu kullanarak tanımlanmıştır.NOT NULL ifadesiyle bu verinin mutlaka dolu olması gerekmektedir.

ENTRY_TIME : tırın giriş zamanıdır TIME olarak tanımlanmıştır. NOT NULL ifadesiyle bu verinin mutlaka dolu olması gerekmektedir.

LEAVE_TIME : tırın çıkış zamanıdır TIME olarak tanımlanmıştır. NOT NULL ifadesiyle bu verinin mutlaka dolu olması gerekmektedir.

ARRIVAL_DATE : tırın fabrikaya giriş yaptığı tarihtir.DATE olarak tanımlanmıştır. NOT NULL ifadesiyle bu verinin mutlaka dolu olması gerekmektedir.

DRIVER_ID : tır sürücüsünün TC kimlik numarasıdır.INT tamsayı olarak tanımlanmıştır ve primary key olarak tanımlanmıştır.Bu da TC kimlik numarasının benzersiz kayıt olduğunu göstermektedir.

EHS_VIDEO : tır sürücüsünün fabrikaya giriş yaptığında fabrikanın kuralı gereği izlemesi gereken iş sağlığı güvenliği videosunu izlemesinden bahseden veridir.BOOLEAN olarak tanımlanmıştır bu da verinin ya “1” ya da “0” olacağını göstermektedir.Video izlendiyse “1” izlenmediyse “0” girilecektir.

PLATE_NO : Plaka numarası TEXT olarak ve veri setinin en önemli verisi olduğu için NOT NULL olarak tanımlanmıştır.

```
1 • use importdb;
2 • CREATE TABLE IF NOT EXISTS SECURITY (
3     DRIVER_NAME VARCHAR(255) NOT NULL ,
4     ENTRY_TIME TIME NOT NULL,
5     LEAVE_TIME TIME NOT NULL,
6     ARRIVAL_DATE DATE NOT NULL ,
7     DRIVER_ID INT PRIMARY KEY ,
8     EHS_VIDEO BOOLEAN NOT NULL DEFAULT FALSE ,
9     PLATE_NO TEXT NOT NULL
10 );
```

Şekil 8 : Güvenlik (Security) tablosunun MySQL kodu

3.3.4 Güvenlik (Security) Tablosuna Veri Girişi

Güvenlik tablosu tanımlandıktan sonra içine verileri INSERT INTO komutunu kullanarak aktarılmıştır.

INSERT INTO SECURTIY ifadesiyle security olarak tanımlanan tablonun içerisine veri ekleme komutu olarak tanımlanmıştır.Sonrasında security tablosuna ait oluşturulan sütunları parantez içinde sırayla belirtilmiştir.

Alt satıra geçirilirken VALUES komutu ile tanımlanan sütunlara sırasıyla ' ', kullanılarak istenilen veriler yazılır.Yazılan verilerden sonra uygun komutlara ve sıraya göre veriler tablonun içine aktarılmıştır.

```
13 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
14 VALUES ('MEHMET','01:00:00','02:00:00','2023-04-03','234567891','1','35ADJ30');
15 INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
16 VALUES ('ALİMEHMET','10:00:00','12:00:00','2023-04-04','234567892','1','35ADJ60');
17 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
18 VALUES ('MUSTAFA','10:10:00','10:40:00','2023-04-05','234567898','0','35ADŞ60');
19 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
20 VALUES ('MERT','10:30:00','11:40:00','2023-04-10','834567898','0','35AIU80');
21 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
22 VALUES ('KEMAL','17:30:00','19:50:00','2023-04-15','154567898','1','35ADJ87');
23 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
24 VALUES ('VELİ','18:45:00','19:20:00','2023-04-16','964567898','0','35ADP04');
25 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
26 VALUES ('KAMİL','19:45:00','21:50:00','2023-05-15','158967898','1','34PLK05');
27 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
28 VALUES ('EŞREF','10:45:00','11:10:00','2023-04-25','158787898','0','35AG2360');
29 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
```

Şekil 9 : MySQL Güvenlik (Security) tablosunun içine veri girme

```

31 VALUES ('CEM','15:45:00','16:10:00','2023-04-20','158000898','1','35AAC684');
32 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
33 VALUES ('CEMAL','14:50:00','15:10:00','2023-05-20','158015898','1','35AAN184');
34 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
35 VALUES ('CEMİL','13:50:00','14:52:00','2023-05-21','758015898','0','35AAU760');
36 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
37 VALUES ('SEMİH','23:50:00','00:05:00','2023-05-25','758016898','0','35AAZ077');
38 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
39 VALUES ('ASIM','00:50:00','03:05:00','2023-05-30','70016898','1','35ABF18');
40 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
41 VALUES ('MUSA','02:30:00','03:05:00','2023-05-31','70006898','1','35ABJ299');
42 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
43 VALUES ('ENES','22:30:00','23:45:00','2023-05-31','70000898','1','35ACA977');
44 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
45 VALUES ('NAİM','02:30:00','03:45:00','2023-05-11','70020898','0','35ABJ74');
46 • INSERT INTO SECURITY(DRIVER_NAME,ENTRY_TIME,LEAVE_TIME,ARRIVAL_DATE,DRIVER_ID,EHS_VIDEO,PLATE_NO)
47 VALUES ('NIHAT','07:30:00','08:45:00','2023-05-11','70025898','1','35ACP63');

```

Şekil 10 : MySQL Güvenlik (Security) tablosunun içine veri girme (devamı)

SELECT*FROM importdb.security komutu ile importdb veri tabanının içinden security adlı tablodan tüm sütunları getir adlı komutu kullanarak verilerin tabloya işleyip işlemediği kontrol edilmiştir.

The screenshot shows a MySQL database interface with a query editor and a result grid. The query executed is `SELECT * FROM importdb.security;`. The result grid displays the following data:

	DRIVER_NAME	ENTRY_TIME	LEAVE_TIME	ARRIVAL_DATE	DRIVER_ID	EHS_VIDEO	PLATE_NO
▶	ENES	22:30:00	23:45:00	2023-05-31	70000898	1	35ACA977
	MUSA	02:30:00	03:05:00	2023-05-31	70006898	1	35ABJ299
	ASIM	00:50:00	03:05:00	2023-05-30	70016898	1	35ABF18
	NAİM	02:30:00	03:45:00	2023-05-11	70020898	0	35ABJ74
	NIHAT	07:30:00	08:45:00	2023-05-11	70025898	1	35ACP63
	Gürkan	11:40:00	11:59:00	2023-01-03	123456747	1	23GK234
	Minever	09:20:00	11:00:00	2023-01-31	123456768	1	23LIM564
	Anıl	10:00:00	10:30:00	2023-02-27	123456788	0	35IK47
	Ali	12:00:00	13:00:00	2023-04-05	123456789	1	35IKL15
	Enes	17:00:00	18:00:00	2023-04-04	123456795	0	34IJ475

Şekil 11 : MySQL Güvenlik (Security) tablosunun çıktısı



Şekil 12 : MySQL importdb ye bağlı tablolar

Böylelikle importdb veri setinin içine import ve security isimli iki tablo verileriyle beraber işlenmiştir.

3.3.5 Genel (Generall) Tablosunun Oluşturulması

Importdb yaratıldıktan sonra “use” komutuyla yaratılan güvenlik tablosundan ve ithalat tablosundan bazı sütunları alarak yeni bir veri seti yaratılmıştır.CREATE TABLE komutunu kullanarak GENERALL adlı tablo oluşmaktadır.GENERALL tablosunda güvenlik tablosundan ve ithalat tablosunda ortak olarak yer alan “PLATE_NO” sütunu ile ilişki kurulup iki tabloda da aynı olan plakaları çeken bir veri seti oluşturulmuştur.

```
1 use importdb;
2 CREATE TABLE IF NOT EXISTS GENERALL (
3   SELECT IMPORT.FORWARDER_NAME , IMPORT.PLATE_NO , SECURITY.ARRIVAL_DATE,SECURITY.EHS_VIDEO,SECURITY.ENTRY_TIME
4   FROM IMPORT LEFT JOIN SECURITY ON SECURITY.PLATE_NO = IMPORT.PLATE_NO );
```

Şekil 13 : Genel (Generall) tablosunun MySQL kodu

SELECT IMPORT.FORWARDER_NAME kullanarak ithalat tablosunun içinde yer alan sevkiyatçının adı seçilmiştir.

SELECT IMPORT.PLATE_NO kullanarak ithalat tablosunun içinde yer alan plaka numarası seçilmiştir.

SELECT SECURITY.ARRIVAL_DATE kullanarak güvenlik tablosunda yer alan tır geliş tarihi seçilmiştir.

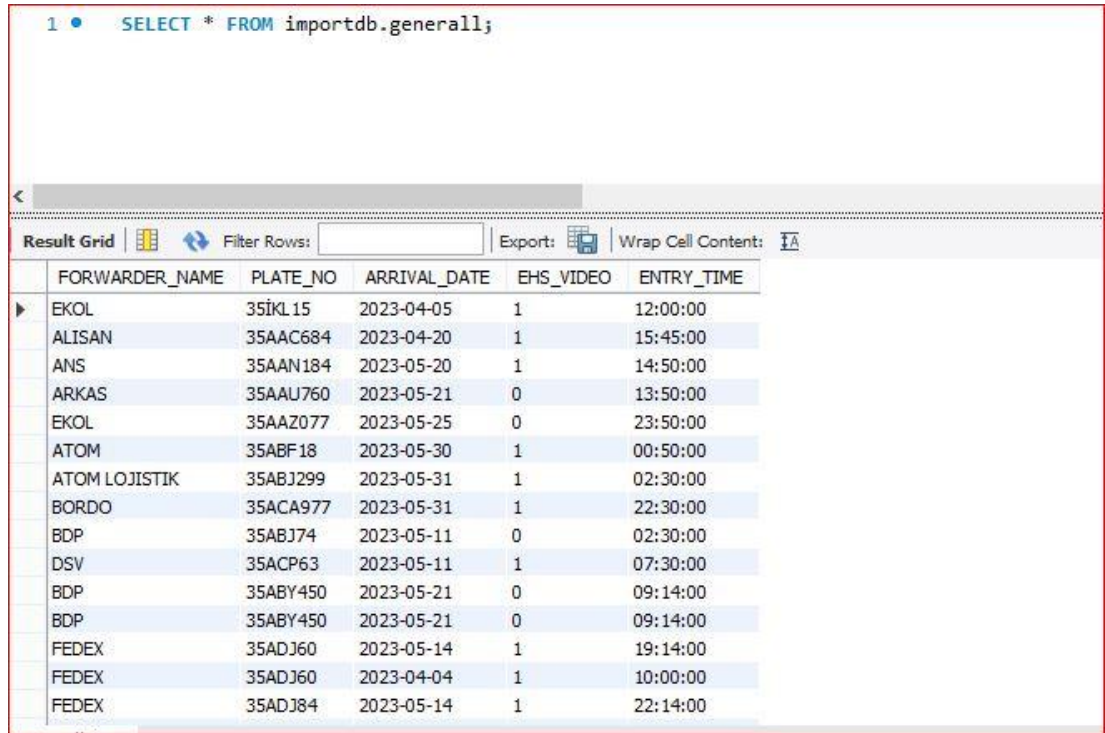
SELECT SECURITY.EHS_VIDEO kullanarak güvenlik tablosunda yer alan iş sağlığı güvenliği videosunun izlenip izlenmediğini gösteren sütun seçilmiştir.

SELECT SECURITY.ENTRY_TIME kullanarak güvenlik tablosunda yer alan tır geliş saatinin verisi seçilmiştir.

İlk satır yazıldıktan sonra FROM kısmında verilerin nereden çekileceği yazılmıştır.

FROM IMPORT LEFT JOIN SECURITY ON SECURITY.PLATE_NO = IMPORT.PLATE_NO

İthalat tablosu ile güvenlik veri tablosunu LEFT JOIN komutunu kullanarak hem ithalat hem de güvenlik tablosunda bulunan verileri aynı zamanda ikisinde ortak olarak bulunan plaka numarasını kullanarak komut yazılmıştır. Komut yazılırken koşul olarak plaka numarasının iki tabloda da aynı olması gerekmektedir.



```
1 • SELECT * FROM importdb.generall;
```

	FORWARDER_NAME	PLATE_NO	ARRIVAL_DATE	EHS_VIDEO	ENTRY_TIME
▶	EKOL	35İKL 15	2023-04-05	1	12:00:00
	ALISAN	35AAC684	2023-04-20	1	15:45:00
	ANS	35AAN184	2023-05-20	1	14:50:00
	ARKAS	35AAU760	2023-05-21	0	13:50:00
	EKOL	35AAZ077	2023-05-25	0	23:50:00
	ATOM	35ABF18	2023-05-30	1	00:50:00
	ATOM LOJISTIK	35ABJ299	2023-05-31	1	02:30:00
	BORDO	35ACA977	2023-05-31	1	22:30:00
	BDP	35ABJ74	2023-05-11	0	02:30:00
	DSV	35ACP63	2023-05-11	1	07:30:00
	BDP	35ABY450	2023-05-21	0	09:14:00
	BDP	35ABY450	2023-05-21	0	09:14:00
	FEDEX	35ADJ60	2023-05-14	1	19:14:00
	FEDEX	35ADJ60	2023-04-04	1	10:00:00
	FEDEX	35ADJ84	2023-05-14	1	22:14:00

Şekil 14 : MySQL Genel (Generall) tablosunun çıktısı

SELECT*FROM importdb.generall komutu ile importdb veri tabanının içinden generall adlı tablodan tüm sütunları getir adlı komutu kullanarak verilerin tabloya işleyip işlemediği kontrol edilmiştir.

Böylelikle oluşan yeni “Genel” tablosuyla hem güvenlik hem de ithalat birimleri ortak bir noktadan bakarak gelen tıra ait en özet en bilinmesi gereken şeyleri bilmesi kolaylaşmıştır.

Güvenlik için gelen kişinin bilgileri ve iş sağlığı güvenliği bilgisi , fabrikaya giren çıkan tırların ve sürücülerinin bilgilerinin bir veri setine kaydedilmesi herhangi bir problemle karşılaşıldığında geriye dönük verilere bakılmasını kolaylaştırmıştır ve kayıt altına alınmıştır.İş sağlığı ve güvenliği video izleme takibiyle eğitimin veri kaydının tutulması kolaylaştırılmıştır.Plaka bilgisinin ortak olması sebebiyle tır giriş çıkışlarında ithalat ekibiyle ortak çalışılması gerektiği önemli hale gelmiştir.

İthalat için tırın içeriye giriş saati , gelme tarihi ve gelen tırın sevkiyatçı bilgisine erişebilir.Belirli tarihte beklenen tırın o tarihte gelip gelmediği , gelmediyse kaç gün sanayi bölgesinde beklediği gibi firmanın maliyetini etkileyecek unsurlar için veri oluşturmuştur.Geriye yönelik yıllık maliyet çalışmaları yapılacağında gerçek anlamda fabrikaya giriş yapmış sevkiyatçılara bakarak ileri yıllara dönük bütçe planlaması yapılabilir.Ya da firmanın tır giriş çıkışları için koyduğu saat dilimini aşan tırlar için yaptırım yapmasına ve fabrika içi lojistiğin düzenlenmesine yardımcı olabilecek bir veri seti oluşmuştur.

Kaynaklar

- [1] Ekinci, S. 2006. <http://www.dokuman.org/?islem=Goster&Dokuman=19>.
- [2] Luke and Welling, Laura Thomson, 2002, PHP and MySQL Web Development, Sams Publishing.

İPEKSU KESKİN

Fabrika Güvenlik Takip Sistemi

YÜKSEK LİSANS PROJESİ

2023