adTempus Database Model

adTempus 4.0

Arcana Development, LLC
www.arcanadev.com/adtempus

Introduction

This document provides an overview of the adTempus database model for users who wish to query the database directly. For more complex programmatic data retrieval, users should consider the adTempus API instead of database queries. For reporting, consider the Report Designer included in the adTempus Console, which allows for custom reports built on top of the API.

Direct updates to the adTempus database are not supported and may result in data corruption. If you need to make programmatic updates to adTempus, use the API.

This document does not provide a data dictionary or a comprehensive entity relation diagram for the database. All table relations are modeled in the database; you can therefore view relations using any reverse-engineering tool or by creating Database Diagrams in SQL Server Management Studio.

All foreign keys link to the OID column in the related table, and the foreign key column names are self-explanatory. For example, in the Job table, the JobGroupOID foreign key identifies the JobGroup that the Job belongs to, and links to the OID column in the JobGroup table.

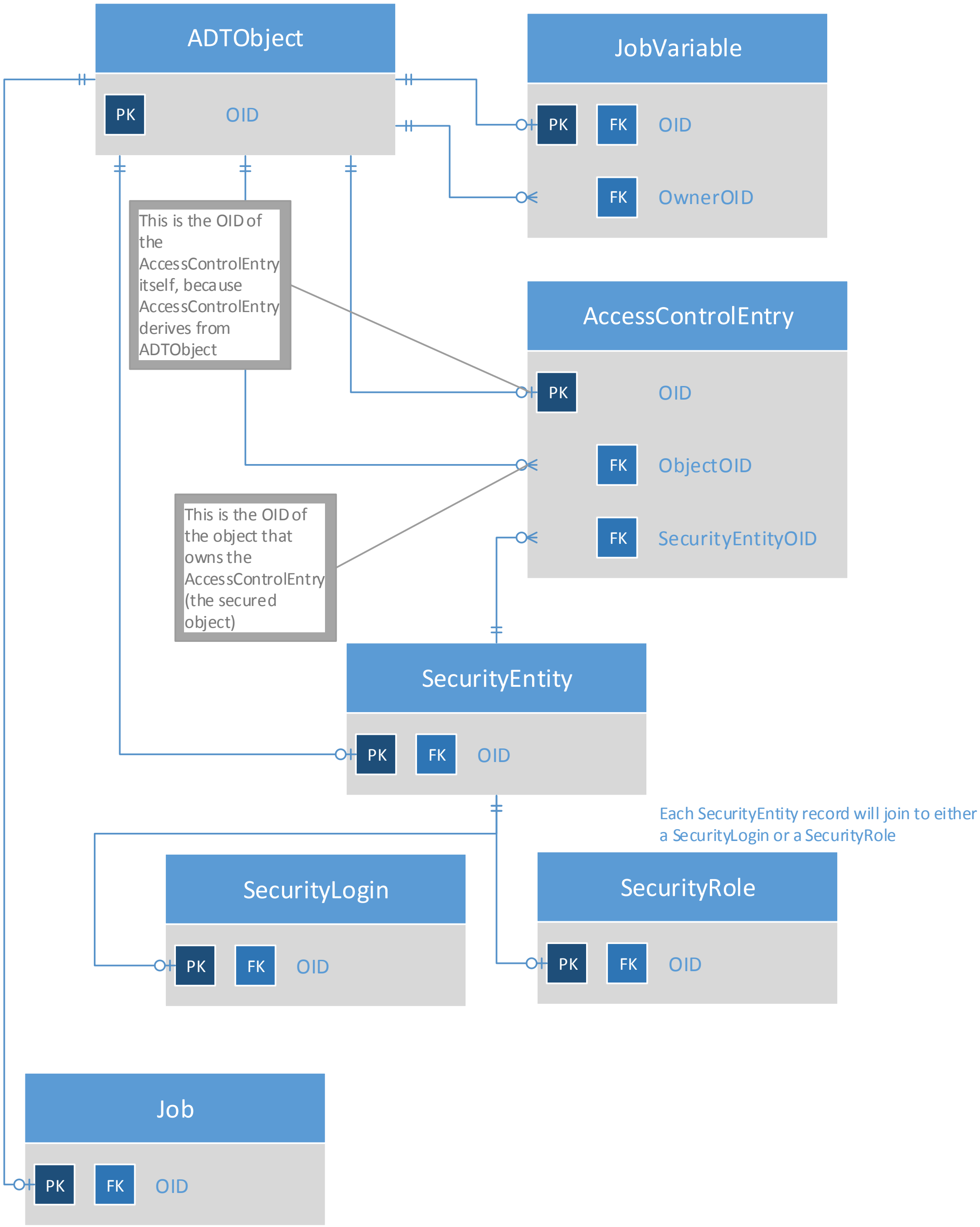
Tables and columns generally map closely to the objects and properties in the adTempus API object model, so refer to the API documentation for details about what tables and columns represent.

Object Model Overview

The database tables reflect the inheritance hierarchy used in the adTempus object model; therefore the properties for a single entity may be split across multiple tables.

Any table where the primary key is named “OID” represents an object derived from ADTObject, and therefore there is a one-to-one relationship to the ADTObject table, using the OID column. ADTObject contains metadata for the object.

For example, the Job inherits from ADTObject. For every row in the Job table, there is a corresponding row in the ADTObject table, using the same OID.



This is the OID of the AccessControlEntry itself, because AccessControlEntry derives from ADTObject

This is the OID of the object that owns the AccessControlEntry (the secured object)

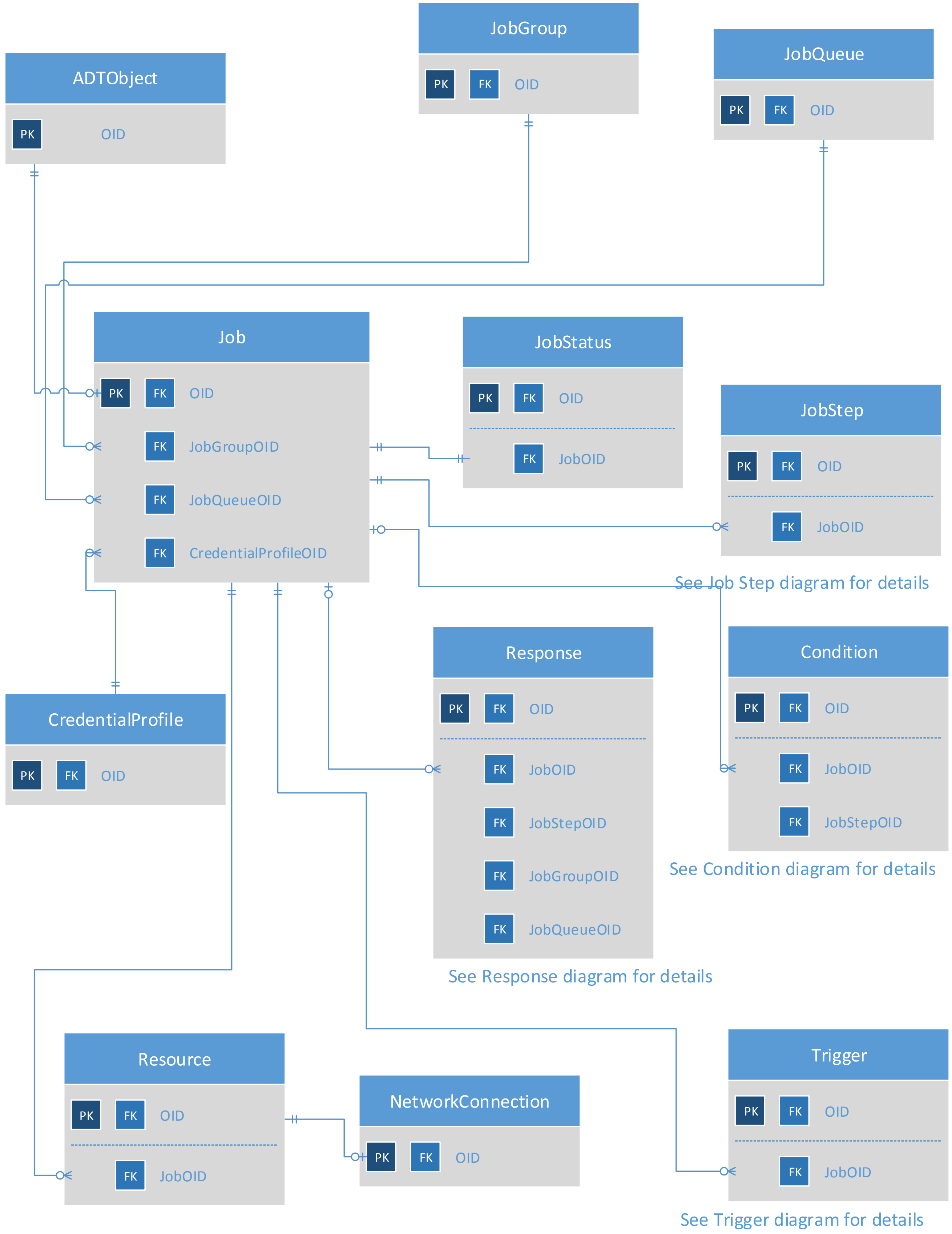
Each SecurityEntity record will join to either a SecurityLogin or a SecurityRole

Depending on the object, there may also be records in the JobVariable and AccessControlEntry tables, as shown above.

To avoid clutter, most diagrams in this document will not show the one-to-one relation with ADTObject.

Job Definition

The following diagram shows an overview of the primary tables that comprise a Job definition. See the following pages for more details on the parts of the job.



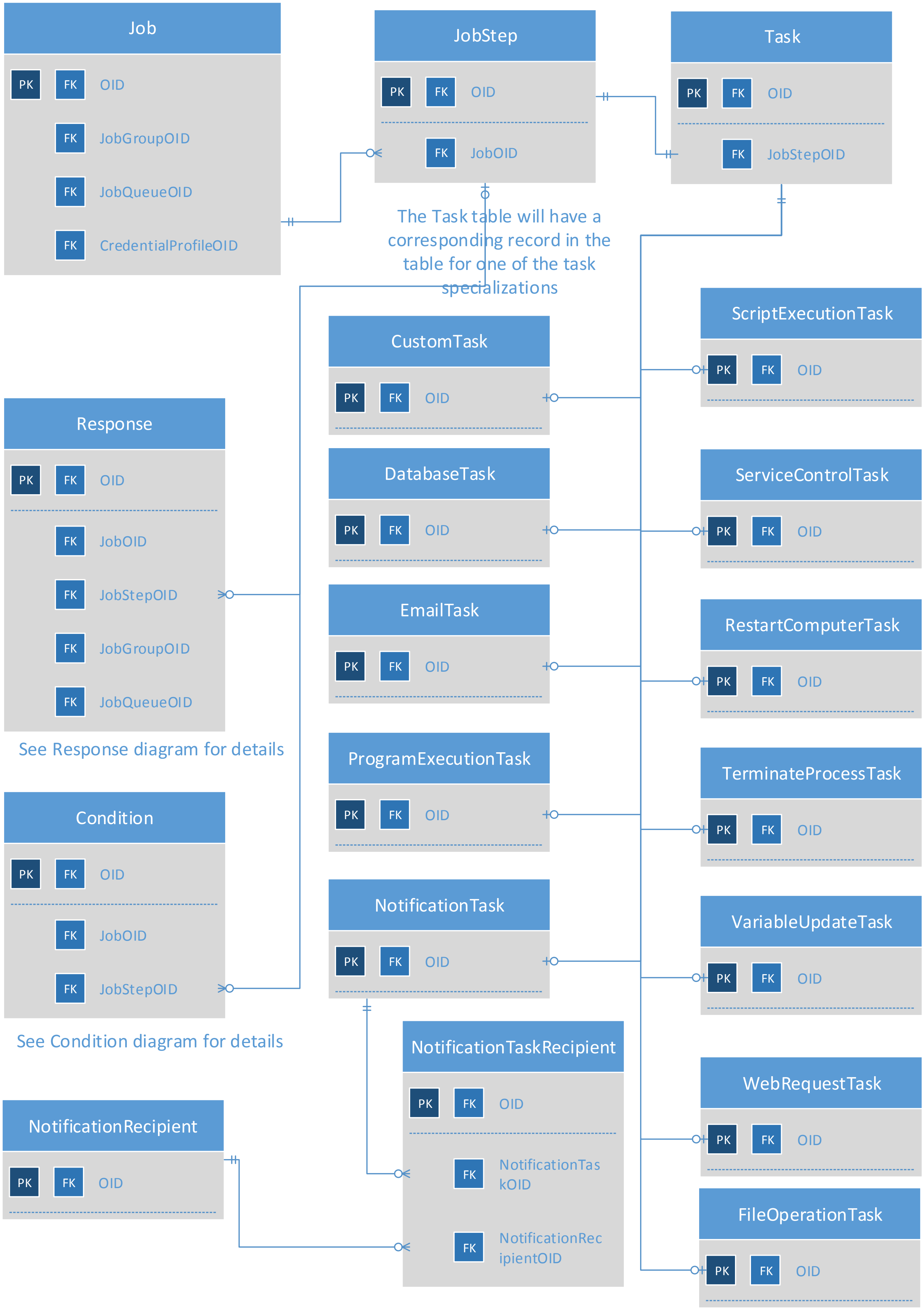
Job Status

The JobStatus table contains information about the state of the job, such as next start time, last start time, status of the most recent instance, etc. For details about specific instances of the job, refer to the Execution History section of this document.

Job Step

Each Job has one or more JobSteps. Each JobStep has exactly one Task.

The Task table will join to a record in one of the specialized task type tables, depending on the kind of task. For example, if the step runs a ProgramExecutionTask the Task table will have a corresponding record in ProgramExecutionTask (with the same OID).



Job
PK FK OID
FK JobGroupOID
FK JobQueueOID
FK CredentialProfileOID

JobStep
PK FK OID
FK JobOID

Task
PK FK OID
FK JobStepOID

The Task table will have a corresponding record in the table for one of the task specializations

Response
PK FK OID
FK JobOID
FK JobStepOID
FK JobGroupOID
FK JobQueueOID

See Response diagram for details

Condition
PK FK OID
FK JobOID
FK JobStepOID

See Condition diagram for details

NotificationRecipient
PK FK OID

CustomTask
PK FK OID

DatabaseTask
PK FK OID

EmailTask
PK FK OID

ProgramExecutionTask
PK FK OID

NotificationTask
PK FK OID

NotificationTaskRecipient
PK FK OID
FK NotificationTaskOID
FK NotificationRecipientOID

ScriptExecutionTask
PK FK OID

ServiceControlTask
PK FK OID

RestartComputerTask
PK FK OID

TerminateProcessTask
PK FK OID

VariableUpdateTask
PK FK OID

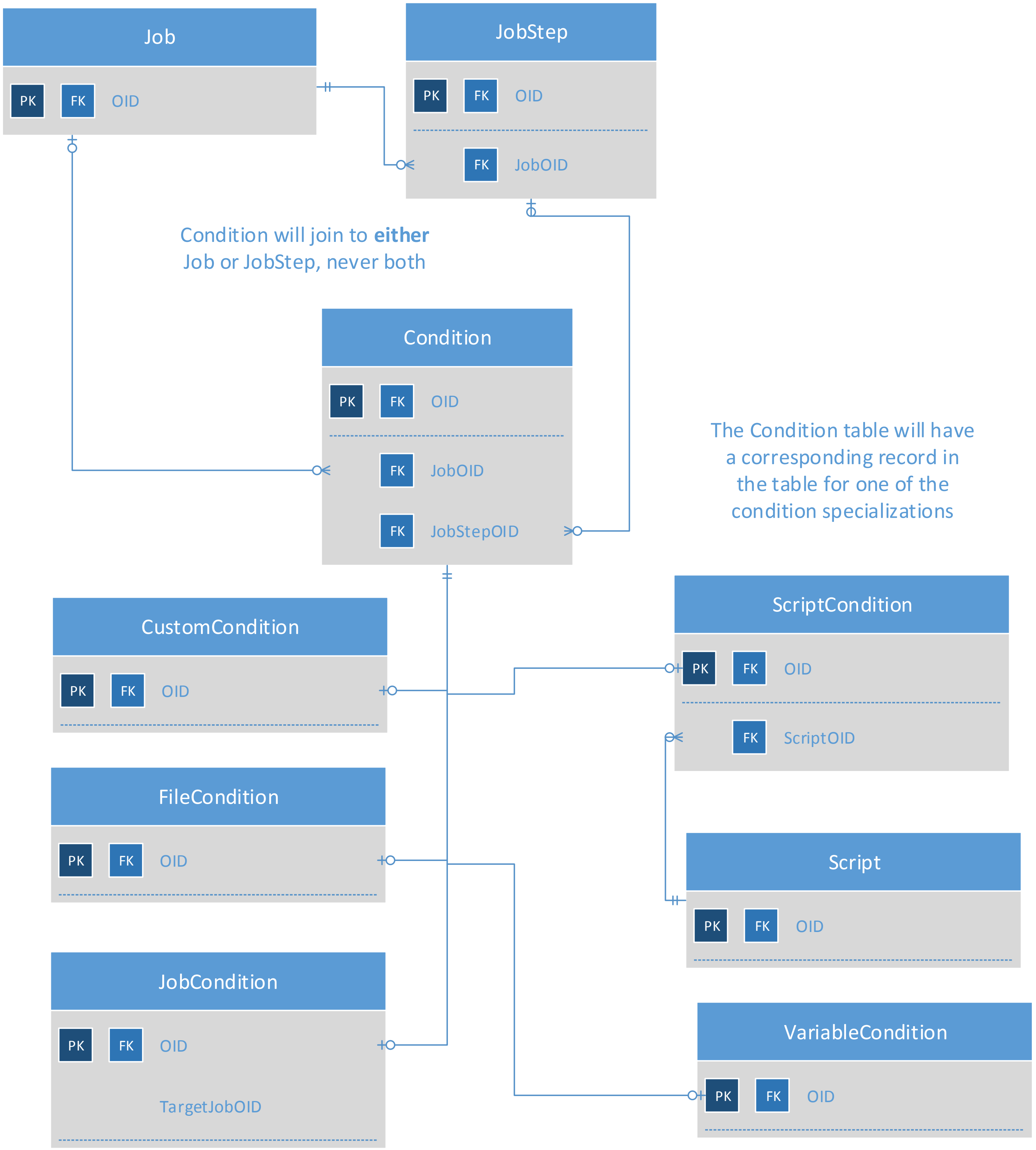
WebRequestTask
PK FK OID

FileOperationTask
PK FK OID

Conditions

Conditions may be attached to Jobs or JobSteps.

The Condition table will join to a record in one of the specialized condition type tables, depending on the kind of condition. For example, if a JobStep has a JobCondition, Condition.JobStepOID will link to JobStep.OID, and Condition.OID will link to JobCondition.OID.



Condition will join to **either** Job or JobStep, never both

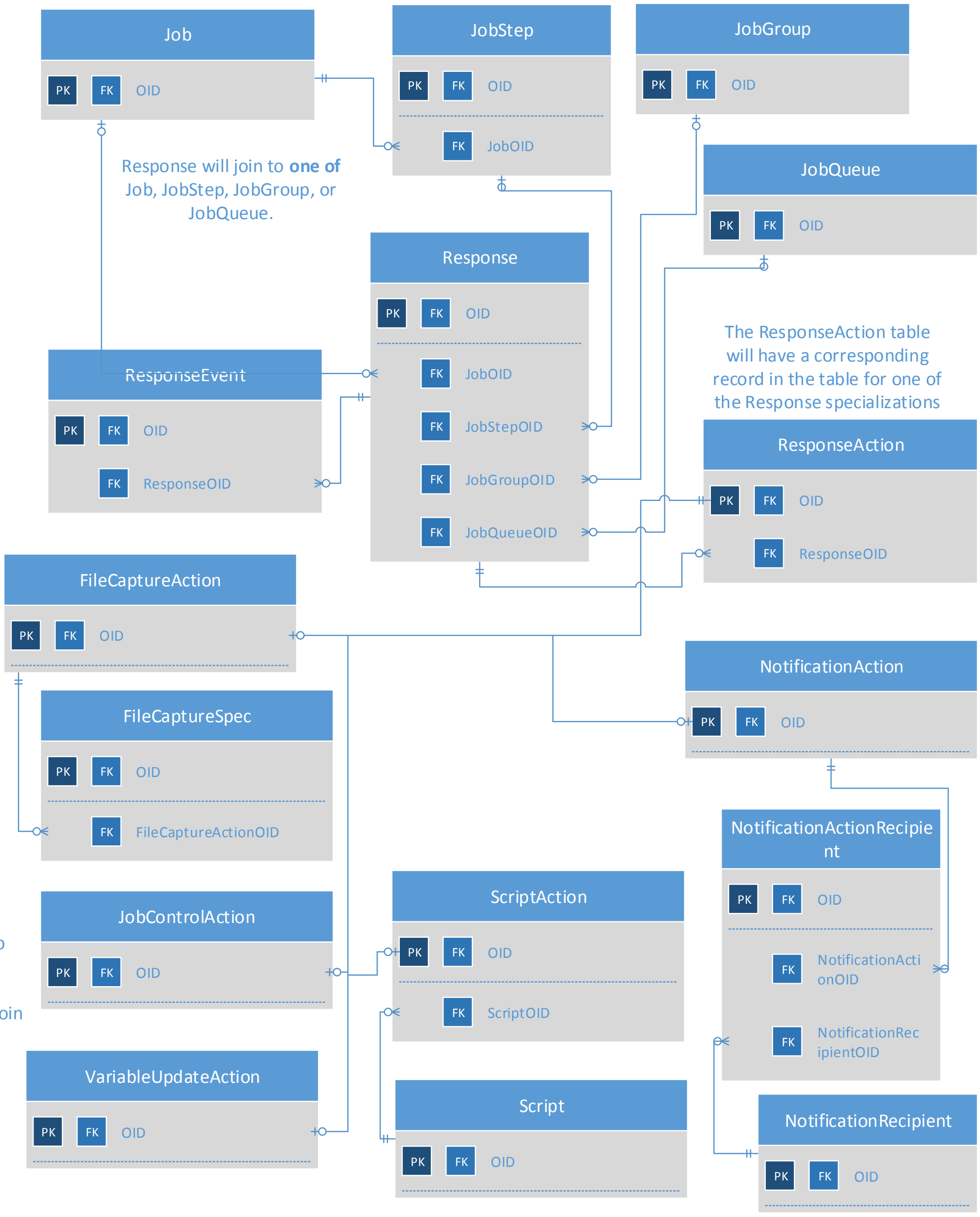
The Condition table will have a corresponding record in the table for one of the condition specializations

JobCondition.TargetJobOID identifies the OID of the job the condition is based on. However, this job may be on a different adTempus server (and thus in a different database) so you cannot always join to the Job table.

Responses

Responses may be attached to Jobs, JobSteps, JobGroups, or JobQueues.

The ResponseAction table will join to a record in one of the specialized action type tables, depending on the kind of action. For example, if a Job has a Response that executes a NotificationAction, Response.JobOID will link to Job.OID, and ResponseAction.OID will link to NotificationAction.OID.



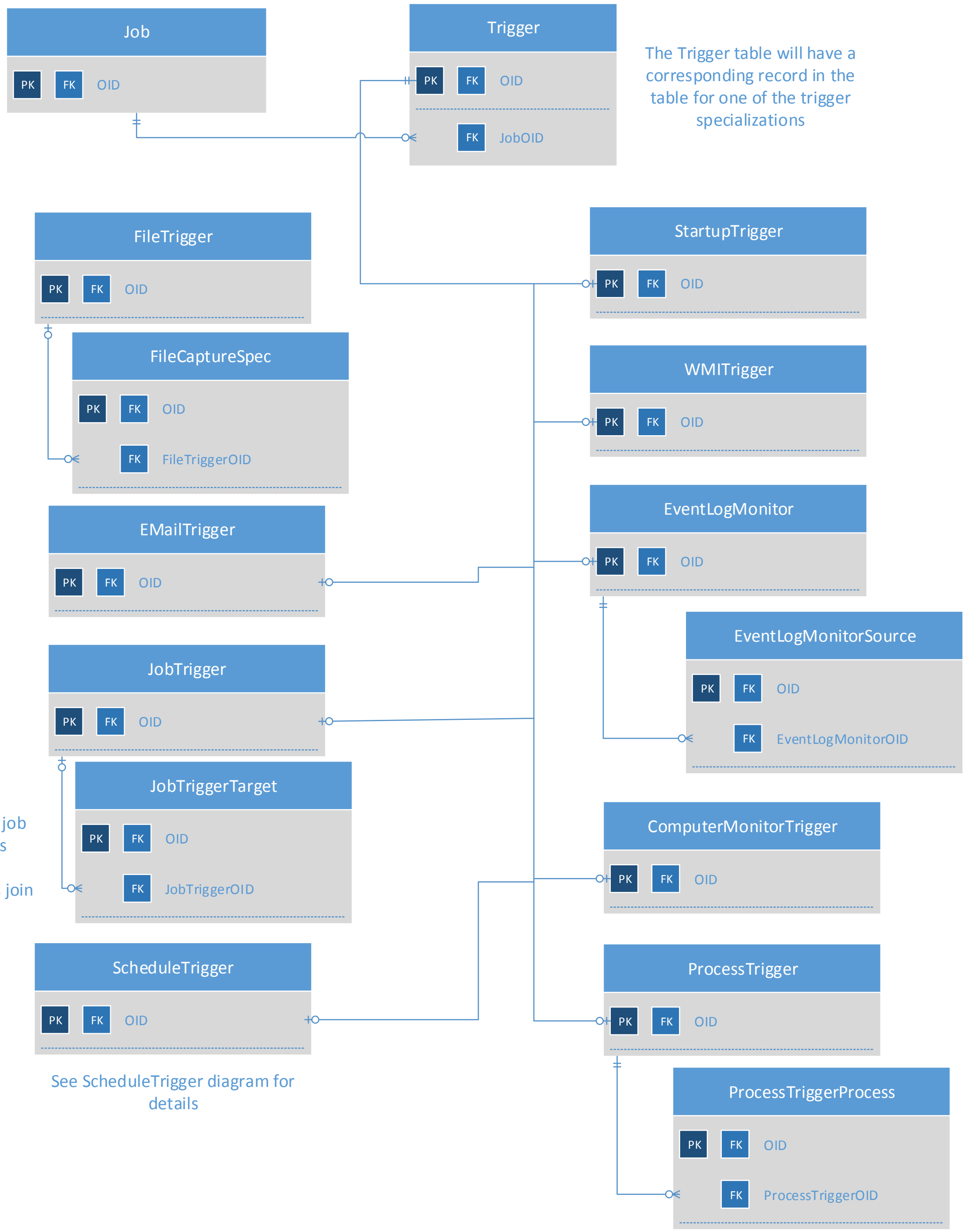
Response will join to **one of** Job, JobStep, JobGroup, or JobQueue.

The ResponseAction table will have a corresponding record in the table for one of the Response specializations

JobControlAction.TargetJobOID identifies the OID of the job the action controls. However, this job may be on a different adTempus server (and thus in a different database) so you cannot always join to the Job table.

Triggers

The Trigger table will join to a record in one of the specialized action type tables, depending on the kind of trigger. For example, if a Job has a ScheduleTrigger, Trigger.OID will link to ScheduleTrigger.OID.



The Trigger table will have a corresponding record in the table for one of the trigger specializations

JobTriggerTarget.TargetJobOID identifies the OID of the job the trigger monitors. However, this job may be on a different adTempus server (and thus in a different database) so you cannot always join to the Job table.

See ScheduleTrigger diagram for details

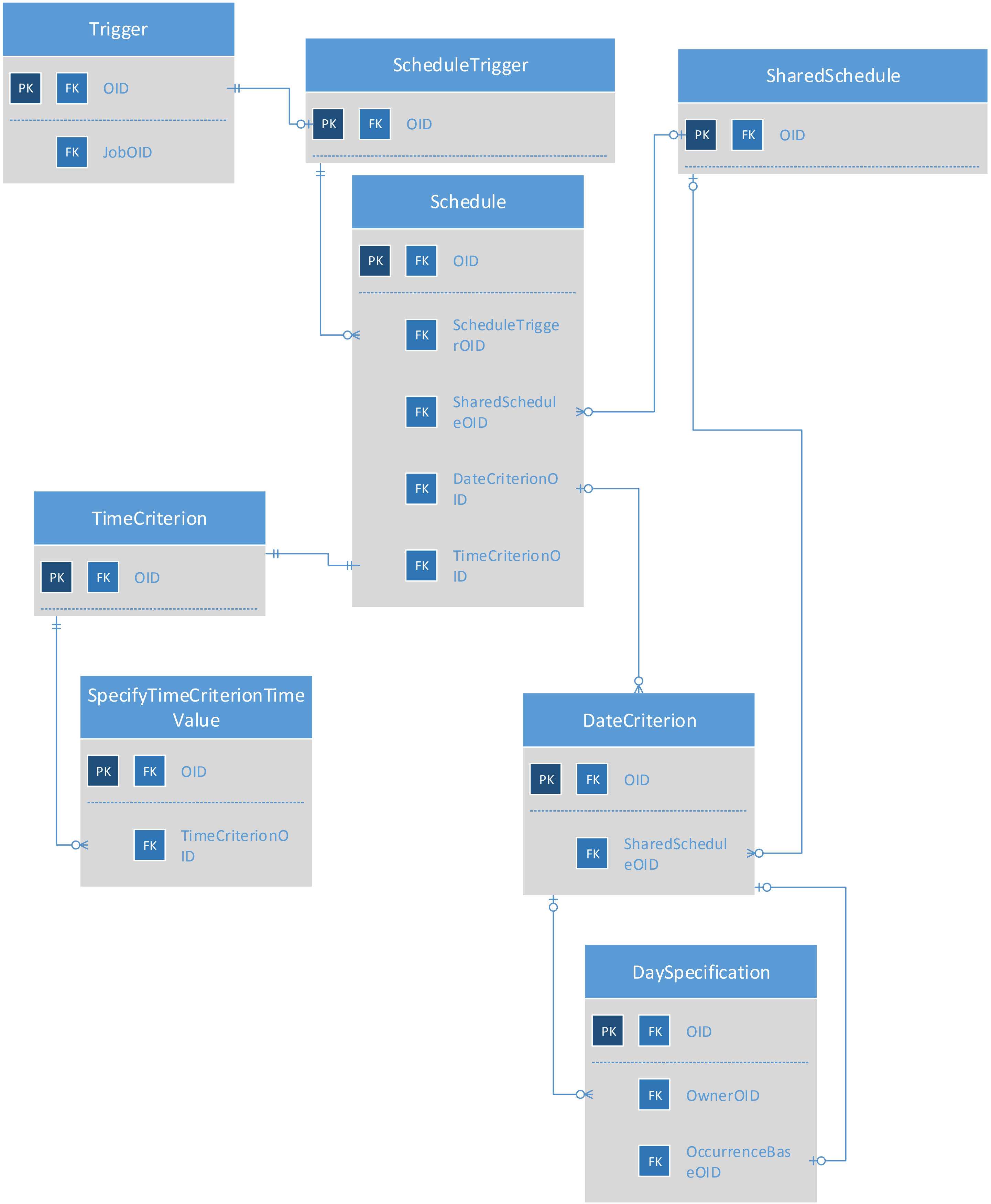
Schedule Trigger

A ScheduleTrigger has one or more Schedules. Each Schedule has a TimeCriterion and either:

- A SharedSchedule, or
- A DateCriterion

A DateCriterion belongs to either a Schedule or a SharedSchedule, never both.

A SharedSchedule may have more than on DateCriterion, to reflect schedules for different years or date ranges.


Trigger
PK
FK
OID
FK
JobOID
ScheduleTrigger
PK
FK
OID
SharedSchedule
PK
FK
OID
Schedule
PK
FK
OID
FK
ScheduleTriggerOID
FK
SharedScheduleOID
FK
DateCriterionOID
FK
TimeCriterionOID
TimeCriterion
PK
FK
OID
SpecifyTimeCriterionTimeValue
PK
FK
OID
FK
TimeCriterionOID
DateCriterion
PK
FK
OID
FK
SharedScheduleOID
DaySpecification
PK
FK
OID
FK
OwnerOID
FK
OccurrenceBaseOID

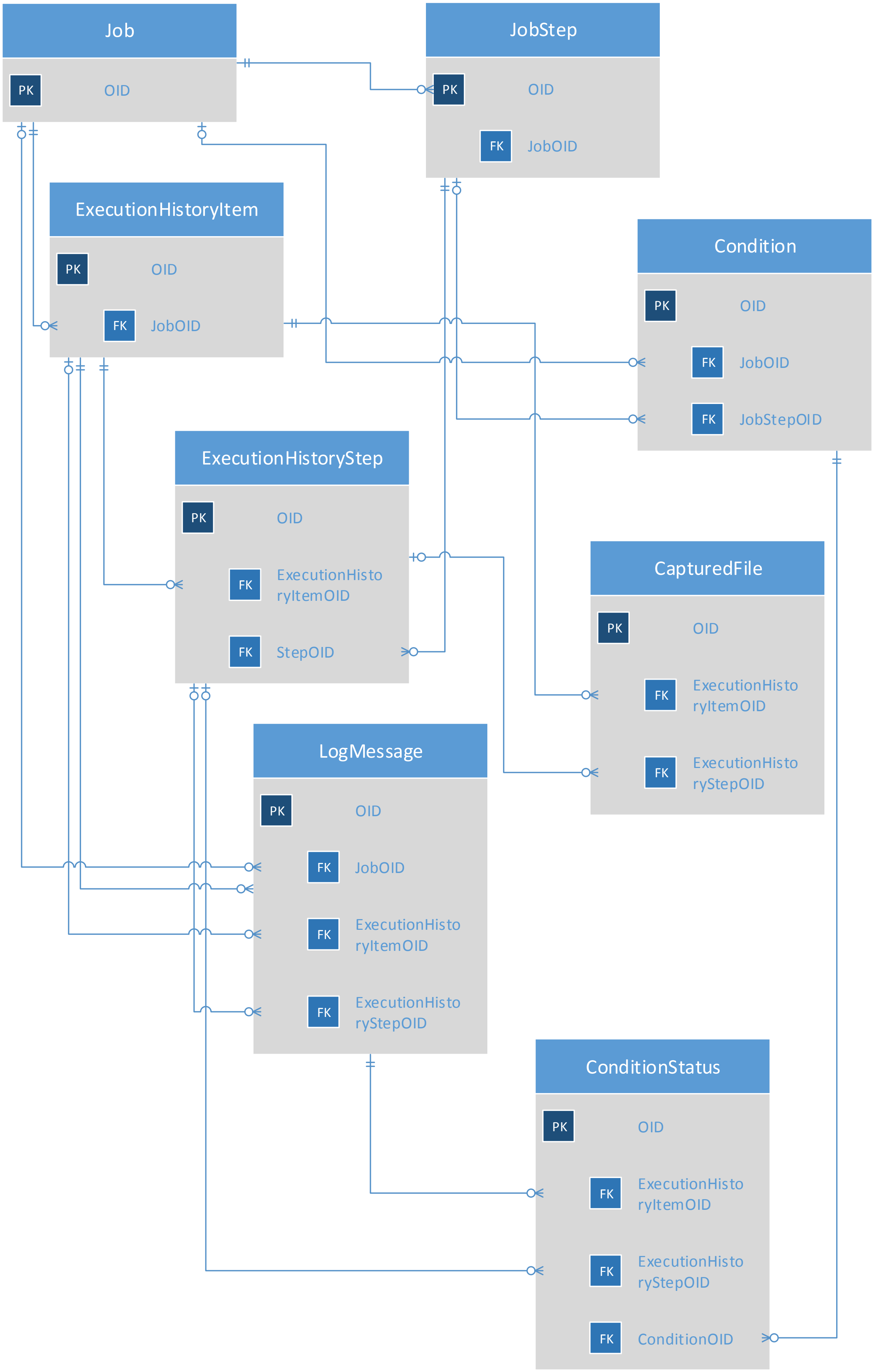
Job History

Each instance (separate execution) of a job is recorded as a record in the ExecutionHistoryItem table. Each ExecutionHistoryItem has corresponding records in ExecutionHistoryStep to record details of each step.

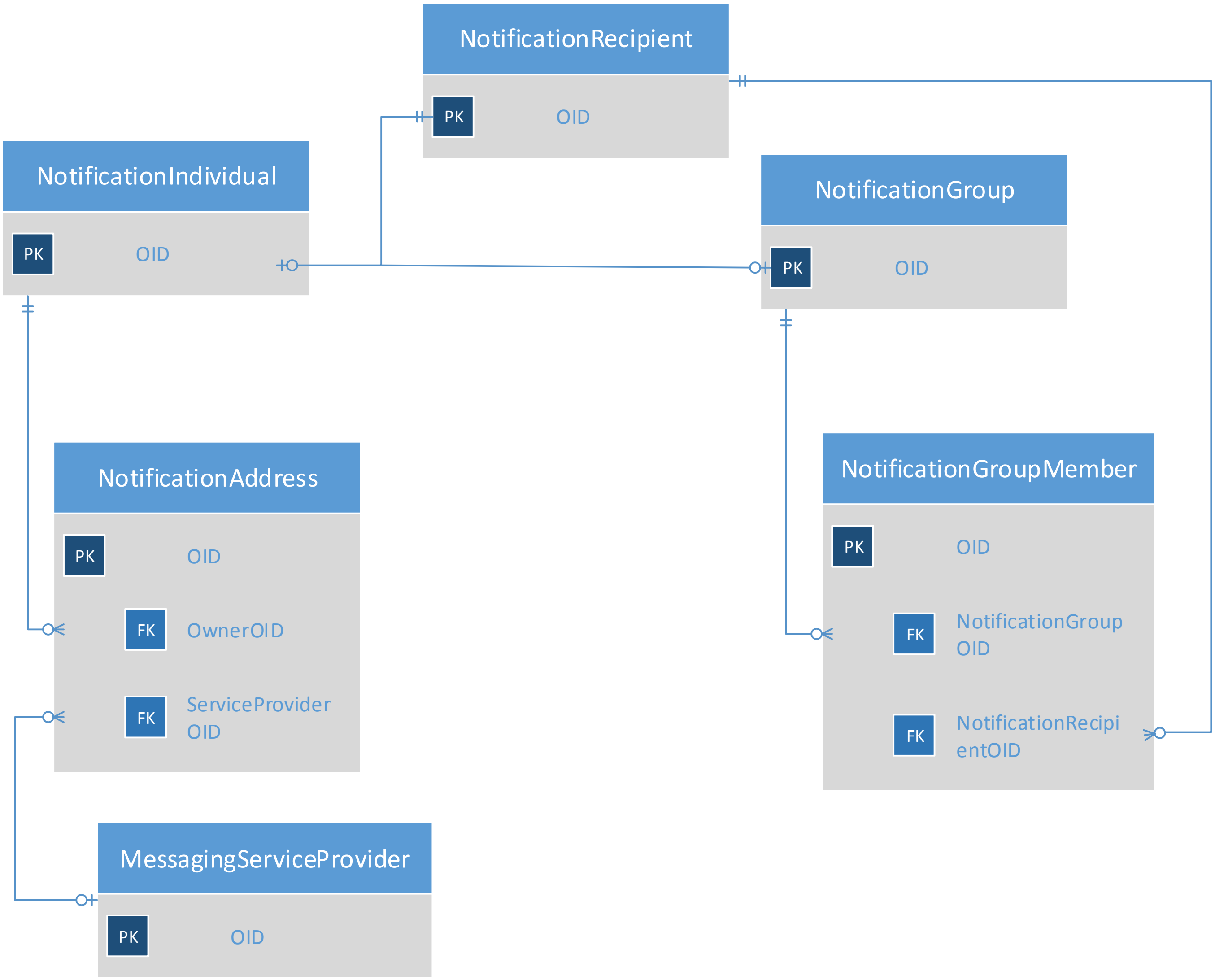
Log messages for the job are found in the LogMessage table. A LogMessage may be owned by:

- Nothing (system-level messages and alerts)
- A Job (JobOID is populated)
- A Job instance (JobOID and ExecutionHistoryItemOID are populated)
- A Job instance step (JobOID, ExecutionHistoryItemOID, and ExecutionHistoryStepOID are populated)

Date/time stamps in the history tables record the date/time in UTC (GMT). For each date/time column there is a corresponding column that records the offset from UTC for the local time on the server at the time the record was created. For example, if a job instance started at 2014-02-15 13:00:00 EST, the ExecutionStart value will be 2014-02-15 18:00:00 and the ExecutionStartTZ value will be -300 (Eastern Standard Time is 300 minutes ahead of UTC). Therefore you can convert a value to local time, you add *ExecutionStartTZ* minutes to the ExecutionStart value.


Job
PK OID
JobStep
PK OID
FK JobOID
ExecutionHistoryItem
PK OID
FK JobOID
Condition
PK OID
FK JobOID
FK JobStepOID
ExecutionHistoryStep
PK OID
FK ExecutionHistoryItemOID
FK StepOID
CapturedFile
PK OID
FK ExecutionHistoryItemOID
FK ExecutionHistoryStepOID
LogMessage
PK OID
FK JobOID
FK ExecutionHistoryItemOID
FK ExecutionHistoryStepOID
ConditionStatus
PK OID
FK ExecutionHistoryItemOID
FK ExecutionHistoryStepOID
FK ConditionOID

Notification Recipient


NotificationRecipient
PK
OID
NotificationIndividual
PK
OID
NotificationGroup
PK
OID
NotificationAddress
PK
OID
FK
OwnerOID
FK
ServiceProvider OID
NotificationGroupMember
PK
OID
FK
NotificationGroup OID
FK
NotificationRecipientOID
MessagingServiceProvider
PK
OID